

Novella Bartolini
Sotiris Nikolettseas
Prasun Sinha
Valeria Cardellini
Anirban Mahanti (Eds.)



22

Quality of Service in Heterogeneous Networks

6th International ICST Conference
on Heterogeneous Networking for Quality, Reliability,
Security and Robustness, QShine 2009
and 3rd International Workshop
on Advanced Architectures and Algorithms for Internet
Delivery and Applications, AAA-IDEA 2009
Las Palmas, Gran Canaria, November 2009, Proceedings



 Springer

Lecture Notes of the Institute
for Computer Sciences, Social-Informatics
and Telecommunications Engineering

22

Editorial Board

Ozgur Akan

Middle East Technical University, Ankara, Turkey

Paolo Bellavista

University of Bologna, Italy

Jiannong Cao

Hong Kong Polytechnic University, Hong Kong

Falko Dressler

University of Erlangen, Germany

Domenico Ferrari

Università Cattolica Piacenza, Italy

Mario Gerla

UCLA, USA

Hisashi Kobayashi

Princeton University, USA

Sergio Palazzo

University of Catania, Italy

Sartaj Sahni

University of Florida, USA

Xuemin (Sherman) Shen

University of Waterloo, Canada

Mircea Stan

University of Virginia, USA

Jia Xiaohua

City University of Hong Kong, Hong Kong

Albert Zomaya

University of Sydney, Australia

Geoffrey Coulson

Lancaster University, UK

Novella Bartolini Sotiris Nikolettseas
Prasun Sinha Valeria Cardellini
Anirban Mahanti (Eds.)

Quality of Service in Heterogeneous Networks

6th International ICST Conference
on Heterogeneous Networking for Quality, Reliability,
Security and Robustness, QShine 2009
and 3rd International Workshop
on Advanced Architectures and Algorithms for Internet
Delivery and Applications, AAA-IDEA 2009
Las Palmas, Gran Canaria, November 23-25, 2009
Proceedings

Volume Editors

Novella Bartolini
Sapienza University of Rome
Computer Science Department, 00198 Rome, Italy
E-mail: novella@di.uniroma1.it

Sotiris Nikolettseas
Computer Technology Institute, Patras University Campus
26504 Rion, Patras, Greece
E-mail: nikole@cti.gr

Prasun Sinha
Ohio State University Columbus, OH 43210, USA
E-mail: prasun@cse.ohio-state.edu

Valeria Cardellini
University of Roma Tor Vergata, 00133 Roma, Italy
E-mail: cardellini@ing.uniroma2.it

Anirban Mahanti
National ICT Australia (NICTA) Eveleigh, NSW 2015, Australia
E-mail: anirban.mahanti@nicta.com.au

Library of Congress Control Number: 2009939478

CR Subject Classification (1998): C.2, H.5.2, C.2.5, G.2.2, H.4.3, H.5.1, H.3.4

ISSN 1867-8211
ISBN-10 3-642-10624-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-10624-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© ICST Institute for Computer Science, Social Informatics and Telecommunications Engineering 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12800926 06/3180 5 4 3 2 1 0

Preface

This volume presents the proceedings of the 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness and of the Third International ICST Workshop on Advanced Architectures and Algorithms for Internet DELivery and Applications.

Both events were held in Las Palmas de Gran Canaria in November 2009. To each of these events is devoted a specific part of the volume.

The first part is dedicated to the proceedings of ICST QShine 2009. The first four chapters deal with new issues concerning the quality of service in IP-based telephony and multimedia.

A second set of four chapters addresses some important research problems in multi-hop wireless networks, with a special emphasis on the problems of routing.

The following three papers deal with recent advances in the field of data management and area coverage in sensor networks, while a fourth set of chapters deals with mobility and context-aware services. The fifth set of chapters contains new works in the area of Internet delivery and switching systems.

The following chapters of the QShine part of the volume are devoted to papers in the areas of resource management in wireless networks, overlay, P2P and SOA architectures. Some works also deal with the optimization of quality of service and energy consumption in WLAN and sensor networks and on the design of a mobility support in mesh networks.

Finally, two sets of chapters are devoted to the problem of data processing and information retrieval in sensor networks, and to the problem of performance optimization in the presence of device heterogeneity in wireless networks.

The second part of the volume contains the proceedings of ICST AAA-IDEA 2009. The first group of chapters in this second part of the volume is devoted to recent advances in the area of networking and in particular to the problem of energy optimization in wired networks, to the design of QoE assessment architectures, and to the issues related to authenticated traffic roaming via tunnels.

The volume closes with a group of chapters dedicated to Web systems and service-oriented architectures, which also includes works on resource management and service selection.

We believe that this volume constitutes a useful resource for both the practitioner and the researcher.

It deals with the complex problem of network heterogeneity and quality of service that is certainly of interest to the practitioner in understanding some practical problems. At the same time, this book is of interest for researchers as it contains a set of contributions that were consciously selected for their novelty and for their research value.

N.Bartolini
S. Nikolettseas

Organization

General Co-chairs

Novella Bartolini
Sotiris Nikolettseas

Sapienza University of Rome, Italy
Patras University, Greece

Steering Committee Co-chairs

Imrich Chlamtac
Xuemin (Sherman) Shen
Xi Zhang

Create-Net, Italy
University of Waterloo, Canada
Texas A&M University, USA

Program Chair

Prasun Sinha

Ohio State University, USA

Poster Chair

Romit Roy Choudhury

Duke University, USA

Invited Session Organizers

Alfredo Cuzzocrea
Hung-Yun Hsieh
Hwangnam Kim
Gianluca Moro
Cigdem Sengul
Ilenia Tinnirello

CNR, Italy
National Taiwan University, Taiwan
Korea University, Korea
University of Bologna, Italy
Deutsche-Telekom Labs, TU-Berlin, Germany
University of Palermo, Italy

Publicity Chair

Simone Silvestri

Sapienza University of Rome, Italy

Publication Chair

Igor Melatti

Sapienza University of Rome, Italy

Local Committee Co-chairs

Alvaro Suarez Sarmiento	University of Las Palmas de Gran Canaria, Spain
Elsa Macias	University of Las Palmas de Gran Canaria, Spain

Workshops Chair

Annalisa Massini	Sapienza University of Rome, Italy
------------------	------------------------------------

Conference Coordinator

Maria Morozova	ICST, Belgium
----------------	---------------

Technical Program Committee

Prithwish Basu	BBN Technologies, USA
Rebecca Braynard	Palor Alto Research Center, USA
Lin Cai	University of Victoria, Canada
Guohong Cao	Pennsylvania State University, USA
Mun-Choon Chan	National University of Singapore
Girish Chandranmenon	Alcotel-Lucent, USA
Shigang Chen	University of Florida, USA
Xiuzhen (Susan) Cheng	George Washington University, USA
Yu Cheng	Illinois Institute of Technology, USA
Yi Cui	Vanderbilt University, USA
Murat Demirbas	University at Buffalo, USA
Do Young Eun	North Carolina State University, USA
Kai-Wei Fan	Cisco Systems
Xiaohui Helen Gu	North Carolina State University, USA
Ting He	IBM T.J. Watson Research Center, USA
Hung-Yun Hsieh	National Taiwan University, Taiwan
Christine Julien	University of Texas at Austin, USA
Koushik Kar	Rensselaer Polytechnic Institute, USA
Hwangnam Kim	Korea University, Korea
Tae-Eun Kim	Extreme Networks
Bong-Jun Ko	IBM T.J. Watson Research Center, USA
Youngbae Ko	Ajou University, Korea
Jeongkeun Lee	HP Labs, USA
Baochun Li	University of Toronto, Canada
Jia-Ru Li	Lilee Systems
Ben Liang	University of Toronto, Canada
Qilian Liang	University of Texas at Arlington, USA
Xiaojun Lin	Purdue University, USA
Thyaga Nandagopal	Bell Labs, Alcatel-Lucent, USA

Jianping Pan	University of Victoria, Canada
Umakishore Ramachandran	Georgia Tech, USA
Violet R. Syrotiuk	Arizona State University, USA
Damla Turgut	University of Florida, USA
Hongyi Wu	University of Louisiana at Lafayette, USA
Zhenyu Yang	Florida International University, USA
Fan Ye	IBM T. J. Watson Research Center, USA
Lei Ying	Iowa State University, USA
Jeonggyun Yu	Samsung Electronics
Hongwei Zhang	Wayne State University, USA
Wensheng Zhang	Iowa State University
Dong Zheng	NextWave Inc.

Table of Contents

QShine 2009: Session I – IP Telephony and Multimedia

QoS Measurement-Based CAC for an IP Telephony System	3
<i>José M^a Saldaña, José I. Aznar, Eduardo Viruete, Julián Fernández-Navajas, and José Ruiz</i>	
Towards Real-Time Stream Quality Prediction: Predicting Video Stream Quality from Partial Stream Information	20
<i>Amy Csizmar Dalal, Emily Kawaler, and Sam Tucker</i>	
Risk-Aware QoP/QoS Optimization for Multimedia Applications in Wireless Networks	34
<i>Yanping Xiao, Chuang Lin, Yixin Jiang, Xiaowen Chu, and Shengling Wang</i>	
COCONET: Co-operative Cache Driven Overlay NETwork for p2p Vod Streaming	52
<i>Abhishek Bhattacharya, Zhenyu Yang, and Deng Pan</i>	

QShine 2009: Session II – Multi-hop Wireless Networks

Opportunistic Multipath Routing in Wireless Mesh Networks	71
<i>Jack W. Tsai and Tim Moors</i>	
Gateways Congestion-Aware Design of Multi-radio Wireless Networks	86
<i>Djohara Benyamina, Abdelhakim Hafid, and Michel Gendreau</i>	
Novel Analytical Delay Model and Burst Assembly Scheme for Wireless Mesh and Optical Burst Switching Convergence	104
<i>Jihene Rezgui, Abdeltouab Belbekkouche, and Abdelhakim Hafid</i>	
Evaluation of a QoS-Aware Protocol with Adaptive Feedback Scheme for Mobile Ad Hoc Networks (Short Paper)	120
<i>Wilder Castellanos, Patricia Acelas, Pau Arce, and Juan C. Guerri</i>	

QShine 2009: Session III – Query and Coverage Issues in Sensor Networks

Adaptive Data Quality for Persistent Queries in Sensor Networks	131
<i>Vasanth Rajamani and Christine Julien</i>	

On-Demand Node Reclamation and Replacement for Guaranteed Area Coverage in Long-Lived Sensor Networks 148
Bin Tong, Zi Li, Guiling Wang, and Wensheng Zhang

Variable Density Deployment and Topology Control for the Solution of the Sink-Hole Problem 167
Novella Bartolini, Tiziana Calamoneri, Annalisa Massini, and Simone Silvestri

QShine 2009: Session IV – Wireless, Mobility, and Context-Aware Services

iDSRT: Integrated Dynamic Soft Real-Time Architecture for Critical Infrastructure Data Delivery over WLAN 185
Hoang Nguyen, Raoul Rivas, and Klara Nahrstedt

Cell Breathing Based on Supply-Demand Model in Overlapping WLAN Cells 203
Shengling Wang, Yong Cui, Ke Xu, Sajal K. Das, Jianping Wu, and Yanping Xiao

Comparative Analysis of QoMIFA and Simple QoS 218
Esam Almasouri, Ali Diab, Andreas Mitschele-Thiel, and Thomas Frenzel

Resource-Optimized Quality-Assured Ambiguous Context Mediation in Pervasive Environments 232
Nirmalya Roy, Christine Julien, and Sajal K. Das

QShine 2009: Session V – Switches, Systems and the Internet

Fluctuations and Lasting Trends of QoS on Intercontinental Links 251
Tomasz Bilski

Performance-Adaptive Prediction-Based Transport Control over Dedicated Links 265
Xukang Lu, Qishi Wu, Nageswara S.V. Rao, and Zongmin Wang

Probabilistic Network Loads with Dependencies and the Effect on Queue Sojourn Times 280
Matthias Ivers and Rolf Ernst

Providing Performance Guarantees for Buffered Crossbar Switches without Speedup 297
Deng Pan, Zhenyu Yang, Kia Makki, and Niki Pissinou

QShine 2009: Invited Session I – Resource Management in Wireless Networks

Joint Optimization of System Lifetime and Network Performance for Real-Time Wireless Sensor Networks	317
<i>Lei Rao, Xue Liu, Jian-Jia Chen, and Wenyu Liu</i>	
Network-Assisted Radio Resource Management for Cell-Edge Performance Enhancement	334
<i>Young-June Choi, Narayan Prasad, and Sampath Rangarajan</i>	
Malicious or Selfish? Analysis of Carrier Sense Misbehavior in IEEE 802.11 WLAN	351
<i>Kyung-Joon Park, Jihyuk Choi, Kyungtae Kang, and Yih-Chun Hu</i>	
Enhanced Bandwidth Allocation for TCP Flows in WiMAX Networks	363
<i>Eun-Chan Park, Chunyu Hu, and Hwangnam Kim</i>	

QShine 2009: Invited Session II – Overlay, P2P Networks and Service Oriented Architectures

A Topologically-Aware Overlay Tree for Efficient and Low-Latency Media Streaming	383
<i>Paris Carbone and Vana Kalogeraki</i>	
Similarity Searching in Structured and Unstructured P2P Networks	400
<i>Vlastislav Dohnal and Pavel Zezula</i>	
Network Attack Detection Based on Peer-to-Peer Clustering of SNMP Data	417
<i>Walter Cerroni, Gabriele Monti, Gianluca Moro, and Marco Ramilli</i>	
A Scalable Approach to QoS-Aware Self-adaption in Service-Oriented Architectures	431
<i>Valeria Cardellini, Emiliano Casalicchio, Vincenzo Grassi, Francesco Lo Presti, and Raffaella Mirandola</i>	

QShine 2009: Invited Session III – QoS and Power Consumption

Throughput and Energy Efficiency in IEEE 802.11 WLANs: Friends or Foes?	451
<i>Pablo Serrano, Albert Banchs, Luca Vollero, and Matthias Hollick</i>	
On the Effects of Transmit Power Control on the Energy Consumption of WiFi Network Cards	463
<i>Francesco Ivan Di Piazza, Stefano Mangione, and Ilenia Tinnirello</i>	

A Novel Power-Efficient Middleware Scheme for Sensor Grid Applications 476
Nikolaos I. Miridakis, Vasileios Giotsas, Dimitrios D. Vergados, and Christos Douligeris

Supporting VoIP Services in IEEE 802.11e WLANs 493
Jeonggyun Yu, Munhwan Choi, Daji Qiao, and Sunghyun Choi

QShine 2009: Invited Session IV – Mobility and QoS Support in Heterogeneous Wireless Mesh Networks

Transparent and Distributed Localization of Mobile Users in Wireless Mesh Networks 513
Mehdi Bezahaf, Luigi Iannone, Marcelo Dias de Amorim, and Serge Fdida

Towards QoS Provisioning in a Heterogeneous Carrier-Grade Wireless Mesh Access Networks Using Unidirectional Overlay Cells 530
M. Kretschmer, C. Niephaus, and G. Ghinea

Integration of OMF-Based Testbeds in a Global-Scale Networking Facility 545
Giovanni Di Stasi, Stefano Avallone, and Roberto Canonico

A Proportionally Fair Centralized Scheduler Supporting Spatial Minislot Reuse for IEEE 802.16 Mesh Networks 556
Parag S. Mogre, Matthias Hollick, Jesús Díaz Gandía, and Ralf Steinmetz

QShine 2009: Invited Session V – Data and Information Processing and Management in Sensor Networks

Cooperative Training in Wireless Sensor and Actor Networks 569
Francesco Betti Sorbelli, Roberto Ciotti, Alfredo Navarra, Cristina M. Pinotti, and Vlady Ravelomanana

Multi-Agent Itinerary Planning for Wireless Sensor Networks 584
Min Chen, Sergio Gonzalez, Yan Zhang, and Victor C.M. Leung

Using Sensor Networks to Measure Intensity in Sporting Activities 598
Mark Roantree, Michael Whelan, Jie Shi, and Niall Moyna

EBC: A Topology Control Algorithm for Achieving High QoS in Sensor Networks 613
Alfredo Cuzzocrea, Dimitrios Katsaros, Yannis Manolopoulos, and Alexis Papadimitriou

Self-organization and Local Learning Methods for Improving the Applicability and Efficiency of Data-Centric Sensor Networks	627
<i>Gabriele Monti and Gianluca Moro</i>	

QShine 2009: Invited Session VI – Performance Optimization and Device Heterogeneity in Wireless Networks

Performance Analysis and Cross Layer Optimization for Multimedia Streaming over Wireless Networks	647
<i>Antonio Ao, Zhung-Han Wu, and Ping-Cheng Yeh</i>	
Credit-Token Based Inter-cell Radio Resource Management: A Game Theoretic Approach	663
<i>Chun-Han Ko and Hung-Yu Wei</i>	
On Using Digital Speech Processing Techniques for Synchronization in Heterogeneous Teleconferencing	679
<i>Hsiao-Pu Lin and Hung-Yun Hsieh</i>	
Interference-Free Coexistence among Heterogenous Devices in the 60 GHz Band	696
<i>Chun-Wei Hsu and Chun-Ting Chou</i>	

AAA-IDEA 2009: Session I - Networking

Optimisation of Power Consumption in Wired Packet Networks	717
<i>Erol Gelenbe and Simone Silvestri</i>	
Revisiting a QoE Assessment Architecture Six Years Later: Lessons Learned and Remaining Challenges	730
<i>Amy Csizmar Dalal</i>	
Efficient Authenticated Wireless Roaming via Tunnels	739
<i>Andreas Noack</i>	

AAA-IDEA 2009: Session II – SOA and Web Systems

Towards the Integration of Distributed Transactional Memories in Application Servers' Clusters	755
<i>Paolo Romano, Nuno Carvalho, Maria Couceiro, Luís Rodrigues, and João Cachopo</i>	
Optimizing Distributed Execution of WS-BPEL Processes in Heterogeneous Computing Environments	770
<i>Qishi Wu, Yi Gu, Liang Bao, Wei Jia, Huichen Dai, and Ping Chen</i>	

Optimal Service Selection Heuristics in Service Oriented Architectures	785
<i>Emiliano Casalicchio, Daniel A. Menascé, Vinod Dubey, and Luca Silvestri</i>	
Feedback-Based Adaptive Resource Control in QoS-Aware SOA Systems with Soft Real-Time Requirements	799
<i>Francisco José Monaco and Pedro Northon Nobile</i>	
Author Index	811

QShine 2009

Session I – IP Telephony and Multimedia

QoS Measurement-Based CAC for an IP Telephony System

José M^a Saldaña, José I. Aznar, Eduardo Viruete, Julián Fernández-Navajas,
and José Ruiz

Communication Technologies Group (GTC) – Aragon Institute of Engineering Research (I3A)
Dpt. IEC. Ada Byron Building. CPS Univ. Zaragoza

50018 Zaragoza, Spain

Tel.: +34 976 762 698

{jsaldana, jiaznar, eviruete, navajas, jruiz}@unizar.es

Abstract. This work presents a Call Admission Control (CAC) system for a SIP-based IP Telephony platform. Configured for a multi-branch enterprise environment, the system enables international calls to be established in two steps: one step using Voice over IP (VoIP) through the Internet between the local office and a VoIP-PSTN gateway placed at destination country, and a second step by means of PSTN, from the gateway to the end-user, accounted with local tariffs. CAC decisions are based on Quality of Service (QoS) measurements, call tariffs and also on the number of available lines in the gateway. The CAC has been implemented within a test platform based on virtualization. Measurements to evaluate and validate CAC's impairment on call establishment delays have been obtained.

Keywords: IP Telephony, VoIP, CAC, MBAC, SIP, virtualization, QoS.

1 Introduction

The use of Internet for voice communications in corporate environments may entail cost savings for enterprises. Conference calls among offices can be carried out using Voice over IP (VoIP) services. For enterprises with offices in several countries, a significant enhancement on communications consists of making use of VoIP services for international conference calls between traditional end-user terminals. These calls can be delivered in two differentiated steps: one step through the Internet by means of VoIP between the enterprise offices, and a second step through the PSTN to reach the end user and accounted with local call tariffs (Fig. 1b).

More specifically, the use of IP Telephony [1] services represents an interesting solution for enterprises since not only implies savings, but also availability and security features.

Users demand a Quality of Service (QoS) similar to the one guaranteed for PSTN. VoIP is a real time service in which the packet delay parameter directly impairs calls' quality. The maximum One Way Delay (OWD) recommended by ITU is 150 ms. [2]. Regarding other QoS parameters, nowadays there exist several solutions for their enhancement related to both control and data planes [3]. At the control plane, the

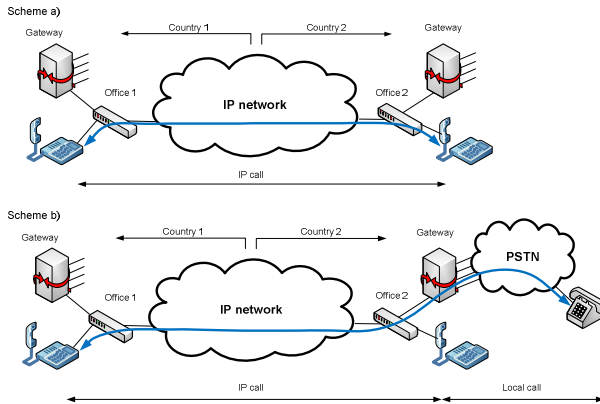


Fig. 1. Traditional and proposed scheme

so-called Call Admission Control (CAC) solution accepts or rejects calls depending on resource availability.

A possible improvement for CAC systems (in terms of QoS and expenses), consists of best route discovery mechanisms for call establishment taking into account that several available locations, from where the call could be established, may exist. To this target, MBAC (Measurement-Based CAC) [4] systems represent a smart option. This work presents a QoS Measurement-Based CAC system developed for a situation in which users have neither control over the Internet, nor over network parameters and the infrastructure. An End-to-End estimation system carries out the measurement of the most relevant QoS parameters over the scenario.

Before setting up an IP Telephony system over the Internet, it is highly recommended to first validate the solution in a controlled environment, such as a testbed platform where several CAC alternatives can be set up and evaluated. Virtualization-based platforms allow deploying a complete network scenario within a single physical machine. Besides, multimedia applications and real protocols can be implemented, achieving realism with low performance cost.

This paper is organized as follows: section 2 discusses the QoS related problems, and their solution using CAC. Section 3 presents the security considerations of this work. System architecture is presented in section 4. The next section covers the test platform. Section 6 presents the validation and preliminary measurements carried out. The last section details the conclusions of the present work.

2 QOS Problem: CAC

Concerning VoIP sphere, there exists a large variety of protocols and configurations for both multimedia data and signaling. The Real Time Protocol (RTP) is usually used for multimedia transmission. At signaling plane, SIP (Session Initiation Protocol) [5], H.323, IAX (Inter-Asterisk eXchange) or MGCP (Media Gateway Control Protocol) are some of the adopted standards. The analysis performed in this work is SIP-based, since it is an open protocol that is widely used in IP networks [6].

Up to now, a complete solution for providing signaling protocols with dynamic configuration and management of QoS parameters does not exist. QoS-guaranteed networks are not prepared to support the massive implementation of multimedia services. Integrated Services (IntServ) [7] mechanisms require all routers along the path to cache signaling information related to each flows, and represents a non scalable solution. On the other hand, Differentiated Services (DiffServ) [8, 9] make use of the Type of Service (ToS) field for traffic classification and also set up a collection of generic rules which establish how each node should react to each traffic flow (Per-Hop Behaviour, PHB). Besides, main problems of this architecture lie in the fact that a mapping process between applications and service classes is first required, and normalising RFC documents still have many open doors to its implementation.

As we previously commented, CAC [10] systems improve call's QoS: They accept or reject new incoming calls depending on the network behaviour. New incoming call acceptance paradigm consists of, while accepting the call, the remaining ongoing calls are not affected in terms of quality, packet losses and delays [11].

A variety of CAC systems have been widely used over several network technologies, such as mobile networks or ATM [12]. Today, in fact, CAC systems constitute a key component in QoS networks defined by NGN (Next Generation network) standardization organisms like 3GPP (*3rd Generation Partnership Project*), WiMAX Forum (*Worldwide Interoperability for Microwave Access*) and TISPAN (*Telecommunications and Internet converged Services and Protocols for Advanced Networking*). Recommendations define a central entity for the management of QoS policies and resources supply. In this scenario, SIP has been adopted by 3GPP as IMS (IP Multimedia Subsystem) [13] signaling protocol.

MBAC represents a suitable CAC option for QoS enhancement. Nowadays, these systems are used in some commercial solutions [14] but their features are limited to manufacturer's devices. For instance, Cisco presents two MBAC SIP compatible systems: AVBO and PSTN Callback [15]. Some other systems are [11]: SU-CAC (Site Utilization-Based CAC), which reserves bandwidth for VoIP communications in the configuration stage, and LU-CAC (Link Utilization-Based CAC), whose decisions are based on host individual bandwidth usage, enabling layer-2 multiplexing, but increasing complexity to the system and using RSVP (Resource ReSerVation Protocol) [16].

The following conditions [12, 17] need to be satisfied for a functional MBAC scheme:

- Ensure that desired QoS level is targeted (precision).
- Maximize resources usage.
- Reach a tradeoff between implementation expenses and revenues.

Recently, different MBAC designs for real time flows, voice and video essentially, have been developed. In [18] it is presented a CAC system that aims to preserve QoS parameters in a wireless mesh network, for a VoIP service. Reference [19] presents a predictive autoregressive CAC algorithm for video distribution systems.

Since MBAC is based on measurements, its implementation requires the usage of estimation and monitoring tools for QoS parameters. There exist several tools (e.g. Nettimer, Pathchar, etc.) which characterize delay, jitter delay, available bandwidth and losses. Depending on the tool, several MBAC systems can be configured. These

tools can be classified in two groups: End-to-End and centralized. The first ones are used to measure parameters at network borders, without considering the inner network structure. On the other hand, centralized tools make use of information obtained from routers' statistics. In case the MBAC system has no control over the network, End-to-End measurement tools are adopted.

Likewise, another classification for QoS measurement tools divide them in active [20] and passive [21] categories. Active tools are based on analyzing the so-called *probe packets*, delivered into the network. Passive ones, capture packets corresponding to network flows for their analysis either online or offline.

3 Security Considerations

One first option to secure users' flows consists of using IPsec at IP level. In fact, enterprises tend to adopt VPN systems for communications among offices. The problem of this solution is that it demands from the VPN-gateways installed in the data centre and offices to be fast enough not to add undesirable delays in voice packets.

Security can also be managed at transport layer. SIP over Transport Layer Security (TLS) protocol is called SIPS. This protocol ensures hop-by-hop security, so that it might only be interesting in case all SIP entities have wholesale relationships and they use PKI (Public Key Infrastructure) for authentication processes. Besides, TLS runs over TCP, while the most suitable protocol for real time traffic is UDP.

Security can also be configured within SIP messages at application layer, as described in RFC 3261: S/MIME authenticates and encodes users' messages. Digest authentication is an accepted method too.

To secure multimedia traffic, SRTP can be used. This protocol provides confidentiality, integrity and authentication to the system and hardly impairs the frame size. AES (Advanced Encryption Standard) is the encoding algorithm for multimedia content. It enables individual frame decoding, a basic feature for real time flows where losses and out of order packet delivery are common.

4 System Architecture

4.1 General Scenario Description

In this article, we implement an End-to-End based MBAC for an IP Telephony system. The initial scheme is similar to the one implemented by Cisco [11]. The IP Telephony system network corresponds to an enterprise with several central offices placed in different countries (Fig. 2). Each office has its own local agent and the PBX is configured within the centralized data centre. To reduce management expenses, it is desirable to keep the dialplan only for the PBX and not to distribute it to the central offices. Furthermore, Internet is used for Telephone traffic delivery among offices, instead of dedicated lines. For a suitable system performance, each office also includes IP-telephones, soft phones, and a VoIP gateway.

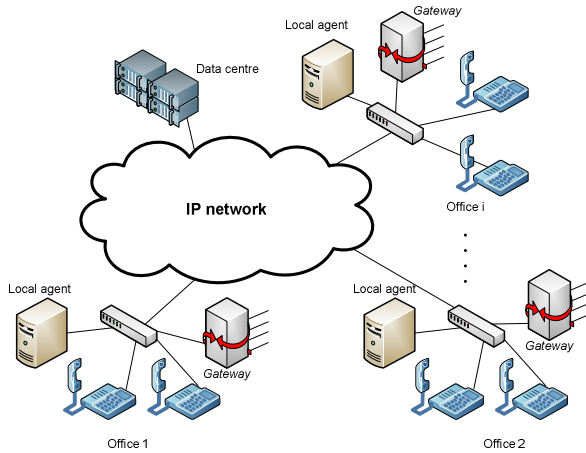


Fig. 2. System architecture

Local agents play a fundamental role in choosing the best route for phone calls in terms of QoS and expenses. First, estimation and monitoring processes of QoS parameters carry out measurements among offices, collecting relevant information for the system. Taking on account this information, the phone tariffs and the number of available and occupied lines in the gateway, the local agent fills in the tables on which the MBAC will base its decisions. Finally, the agent, through its SIP proxy, processes the phone calls signaling to implement the CAC mechanism, based on the information provided by the tables.

This work relies on the following assumptions:

- 1) The measurement system has been designed to dynamically adapt to connection characteristics with a tight inter-estimation time. A high inter-estimation time would derive a lack of accuracy in the measurements, and a short one may result in network overloading.
- 2) The data centre comprises a high availability system, security, back-up files and other functionalities to ensure a proper operational behavior without interruptions. A broadband Internet connection with enough bandwidth is also available.
- 3) There exists a function that calculates and takes the CAC decision based on QoS measurements, accounting tariffs and lines' occupancy in gateways.

4.2 CAC System Operation

As it has been depicted, the system includes a single PBX that includes the dial plan. There is a local agent in each central-office in charge of signaling, configured in such a way that all signaling messages among the PBX and the terminals will pass through it. Thus, signaling information can be sent to the CAC in order to take decisions about future connection requests and keep count of the number of calls established in the

gateway at any time. In case no call rejections are notified, the agent only retransmits signaling messages.

Internal office calls are directly managed by the local agent and do not require PBX functions to be established. For cases in which calls between offices do not go through the PSTN, CAC decisions are just for call acceptance or rejection, since it would be worthless to redirect them to an office different from the destination one.

In order to adopt CAC decisions, each local agent makes use of a so-called “decision table” (Table 1), in which it is specified how the agent may act in case a call request (INVITE) SIP message is received.

Table 1. Decision table

Origin	Internal call	call to PSTN
1	Accept / reject	Accept / reject / redirect to <i>i</i>
2	Accept / reject	Accept / reject / redirect to <i>i</i>
...
N	Accept / reject	Accept / reject / redirect to <i>i</i>

This table is built from other tables discussed in next sections, which depend on QoS parameter estimations, telephone tariffs and number of available lines in the gateway.

$$Decision\ table = f(QoS\ measures,\ tariffs,\ available\ lines) \tag{1}$$

When a local agent receives an INVITE from the PBX (Fig. 3), sent to a terminal, it accepts or rejects it, depending on the corresponding record filled in the table. The incoming call can be rejected due to lack of QoS in the route between source and destination offices, or due to unavailability of destination terminal.

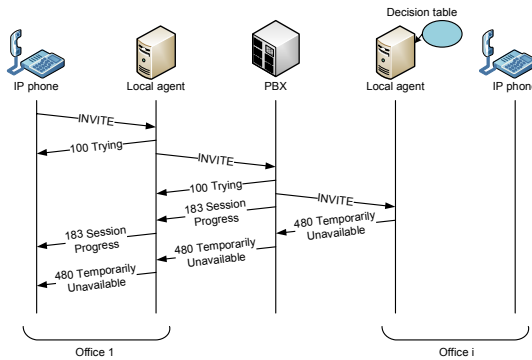


Fig. 3. Rejected call

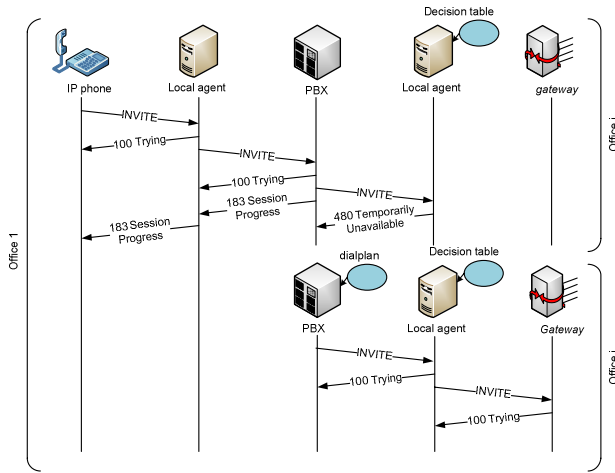


Fig. 4. PBX tries to establish a call by the gateway of another office

In case the local agent receives an INVITE message sent to the gateway and the table indicates that it must be rejected, a “480 Temporarily Unavailable” SIP message (Fig. 4) will be sent to the PBX. The PBX, according to its dial plan, may try to establish the call through other office’s gateway with an economic tariff for the destination of the call.

In case the table indicates “redirect”, the agent acts as *redirect server*, re-routing the call to the central office which owns available lines to establish the connection (Fig 5). The *redirect server* sends a 3XX message reporting about an alternative route.

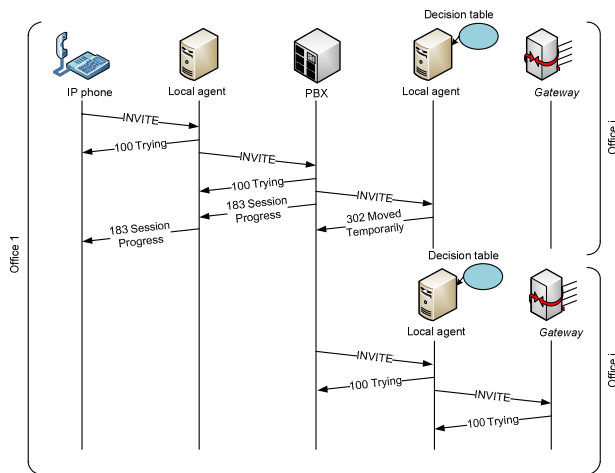


Fig. 5. A call is redirected to a new office

Then, the PBX tries to reach the new destination without consulting its dial plan. The agent who acts as *redirect server* will not take part in this call again.

4.3 Control Tables

As we have previously indicated, each agent's "decision table" is filled in from other tables that are next described:

1) QoS table (Table 2): In each office, a table containing QoS parameter estimations is configured. This table is not mandatory to be symmetric, since there exist broadband access networks that neither are. Elements belonging to the diagonal are not defined since they would represent measurements inside an office. Depending on the measurements considered, it may exist several tables with different QoS parameters.

Table 2. QoS table

	Agent 1	Agent 2	...	Agent i
Agent 1	-	Par 1→ 2	...	Par 1→ i
Agent 2	Par 2→1	-	...	Par 2→ i
...
Agent i	Par i→ 1	Par i→ 2	...	-

2) Tariff table (Table 3): Each field is an integer number representing a tariff rate, corresponding to a call from an origin office gateway i to a destination country j . Local calls are designed as "level 1" and the level increases as the price does.

Table 3. Tariff table

	Country 1	Country 2	...	Country j
Gateway 1	Tariff 1→1	Tariff 1→ 2	...	Tariff 1→ j
Gateway 2	Tariff 2→ 1	Tariff 2→2	...	Tariff 2→ j
...
Gateway i	Tariff i→ 1	Tariff i→ 2	...	Tariff i→ j

3) Gateway line table (Table 4): This table lets the system know PSTN lines available at each office. First column indicates the total number of lines in the gateway. The second one indicates the number of occupied lines.

Table 4. Gateway line table

	Total lines	Busy lines
Gateway 1	TL 1	LO 1
Gateway 2	TL 2	LO 2
...
Gateway i	TL i	LO i

The table of decisions (Table 1) is filled in from *QoS*, *tariffs* and *gateway line* tables by using (1). This table can be set up from the information of a single office or taking on account the measurements obtained from all offices.

In case each local agent only knew about its own office parameters related to the rest of them, Table 2 would result in a single column array. *Gateway line table* (Table 4) would be minimized to a counter of available free lines within the office itself. Thus, there would not be information exchange (QoS, lines and tariffs) with the remaining offices. The *Tariff table* could be completely filled in for all offices beforehand, since its refreshing period is large enough (daily or weekly).

In case local agents have the possibility of collecting information concerning all other offices, the function in (1) may have added complexity. If tables' processing time highly increases, (1) might have no validity, in case it runs too long from the instant when measurements were carried out.

The SIP proxy integrated within the agents may act as a *redirect server*, re-routing connections (Fig. 5) towards the office which presents the best conditions to establish the call. To this target, it is necessary to have information about all other offices; otherwise there could appear undesirable effects (e.g. signaling loops).

5 Test Platform

In this chapter we will present the test platform in which the system has been implemented. The design has to adapt well to the IP Telephony system, emulating realistic conditions, and allowing tests and measurements with flexibility.

5.1 Simulation, Real Environment, or Virtual Emulation

To build up the platform there are several options to be considered. Simulation tools are one of these options. In fact, they have already been used in other CAC systems studies [22, 23]. There exist several simulation tools (OPNET, OMNET++, NS-2), nevertheless, they do not implement concrete protocols deployed over a real scenario. The testbed platform could be also be implemented with real devices. However, it may result in high hardware costs, due to the large number of devices that make up the scenario.

Some studies [24, 25] make use of several machines within an only physical computer in order to minimize costs and optimize the testbed control. For instance, User Mode Linux (UML) has been used in the implementation of some emulators like vBET [26]. Virtual nodes connect to each other through an emulated network running under the network card driver. This concept matches with the test environment we want to deploy.

5.2 Test Platform Requirements

The most valuable requirement is to achieve a realistic test platform at a low cost. By using virtual machines, we are able to make use of real applications and complete protocol stacks in their most suitable versions.

Regarding scalability, network configuration enables the platform to be extended with physical machines in case computation and calculation needs became higher due to traffic management or the number of virtual nodes taking part in the test.

Additionally, all tests are being carried out within the same physical machine, guaranteeing test repeatability, since we have configured an isolated and controllable platform.

5.3 Selection of Virtualization Technology

Virtualization consists of using a set of machines, each of them with its own OS, which are executed over the real hardware of a single physical machine.

Considering all virtualization schemes that can be distinguished [27], we have selected one designed as *paravirtualization*, which includes within the client OS the required modifications to avoid any instruction to be managed with privileges. This requirement let the environment run with an execution speed close to non virtualized schemes. Besides, several machines can simultaneously run. The solution we have adopted is based on XEN *paravirtualization*.

Comparative virtualization platform studies [28] have shown XEN as a suitable tool in terms of overhead, linearity and isolation among virtual machines. Communication performance for a scenario composed of 10 virtual machines has been measured in 93MB/s between pairs.

These characteristics are highly interesting for the platform performance, since it is desirable to have a controlled environment in which all virtual machines share available resources in an equitable way.

5.4 Physical Machine Features

The machine within the platform has been developed works based on CentOS 5 OS. Linux core version is 2.6.18-8.1.15. It has a Core 2 Duo at 2.40 GHz processor, 2MB of Cache level 2, and 4GB RAM. Virtual machines also work with CentOS 5. The version of Xen is 3.03-25.0.4.

CPU usage has been monitored during the tests, in order to avoid the influence of processor load on measurements. Utilization has never exceeded 10%.

5.5 System Implementation in the Test Platform

The devices chosen for the developed scenario should require low computational load, due to the fact that they run within a virtualized environment. The required components comprise a PBX, soft phones, SIP proxy servers and VoIP gateways. Only free software solutions have been used.

5.5.1 PBX

Asterisk 1.6, a software PBX developed by Digium, has been configured. Asterisk represents an interesting solution because of its flexibility, updates and GNU-GPL license distribution. It supports SIP, H.323, IAX and MGCP signaling protocols.

A dial plan has been used to permit the redirection of an incoming call to another office in case the gateway chosen as first option does not accept it.

If a scenario whose PBX can be tuned is considered, it must be taken into account that Asterisk includes the so-called *Asterisk RealTime* (ARA) [29] tool, which offers a method to cache and save configuration files in a MySQL or PostgreSQL database. *Static* mode requires a *reload* each time a change in the PBX is carried out. On the other hand, the *dynamic* mode allows Asterisk accessing the database and updating the configuration files in real time. In this case, the dial plan is also dynamically updated according to the QoS parameters measured by agents, gateway availability lines and accounting tariffs.

5.5.2 SIP Proxy

Proxy requirements may include *redirect server* option and the possibility to be tunable so that CAC decisions can be implemented. It must be capable of accessing external information placed at a database too.

The OpenSIPS 1.4 free software version has been the selected solution. It has *register server*, *location server*, *proxy server* and *redirect server* functionalities. Low computational load and the possibility to add and delete functionalities in a modular way are also smart features. At transport layer it supports UDP, TCP, TLS and SCTP. At network layer both IPv4 and IPv6 are supported.

SIP proxy configuration is done with a high level programming language, within the *opensips.cfg* in which it is specified what the proxy should do for each received message. MySQL access is available too.

AAA (Authentication Authorization and Accounting) functions can be managed through databases (MySQL, PostgreSQL or text files), RADIUS or DIAMETER protocols.

5.5.3 Soft Phone

PJSUA 1.0 is the soft phone implemented at local offices. It is part of the PJSIP project. This project offers a complete SIP stack under the GPL license. PJSUA is the reference command line soft phone utilized in PJSIP to achieve the whole SIP protocol implementation and its footprint is smaller than 150KB. It supports simultaneous calls, call waiting and voice messages functionalities, UDP, TCP, TLS and SRTP protocols and Speex, iLBC, GSM, G711 and G722 codecs. Finally, there are also available NAT functionalities (ICE and STUN).

5.5.4 Gateway

IP Telephony gateways are also necessary to complete the emulated platform. It must also be taken on account that PSTN connections are also emulated.

The solution adopted makes use of the PJSUA soft phone, since it supports simultaneous calls, this way limiting the number of gateway lines to emulate. Thus, whenever all PJSUA lines are occupied, the system will consider that the gateway has run out of lines and it will reject future calls.

5.6 System Security

Since the measurement plan still remains in its early stage, security protocols have not yet been included. We next enumerate considered possibilities.

In case IPsec tunneling was available among the data centre and each office for any traffic flow, it could also be used for voice traffic without requiring security protocols in upper levels. Nevertheless, due to the fact that voice is a real time application, upper layer protocols could in certain situations confer better performance and offer better QoS than IPsec.

SIP signaling can be guaranteed through TLS. The main drawback of this solution lies in the need of having one TLS tunnel per existing link. Besides, TLS works over TCP, and additional delay and overhead are susceptible to impair the system.

Asterisk does not natively support SRTP. Thus, it has been tested to directly send SRTP traffic between two soft phones without going through the PBX. The main issue lies in the fact that the system can only run in this configuration mode when both end user firewalls manage voice traffic. On the other hand, the traffic should also pass through the PBX. In this case TLS should also be used for RTP flows. SIP proxies do not take part in multimedia flow management, since they are only in charge of signaling at the control plane.

Implementing the CAC in the system implies the addition of local agents to the traditional scenario which only includes the PBX and the soft phones. If TLS is the solution adopted, the appropriate certificates must be used for each one of the four TLS tunnels: Between each soft phone and its proxy and between the proxies and the PBX. To configure this TLS scenario, TCP transport for SIP must also be supported within Asterisk. This feature has been already included in Asterisk 1.6. version.

6 Validation and Preliminary Measures

The CAC system validation has been carried out over SIP in the test platform, composed of virtual machines with three central offices apart from the PBX. For each test, we have performed 10 measurements to obtain the mean and standard deviation values. These measurements have been performed independently of the transmission and propagation times, since they are not yet implemented in our emulated environment. Thus, we talk about processing time measurements.

We measure T_{pnoCAC} (machines' processing time without the CAC system) and T_{pCAC} (processing time with CAC). Furthermore, in the emulated scenario, obtained measurements will be impaired due to the fact that all the machines are running over the same hardware; thus, obtained delay measurements represent an upper limit of the

processing time in real machines. Besides, we define T_{UL} as the propagation delay at the upload link and T_{DL} the propagation delay at the download link (Fig. 6).

Including the local agents entails two new machines in the route of signaling packets to go through. This implies adding two times the delay produced at the local network (Fig. 6 and 7). This delay is negligible compared to Internet propagation delays.

We want to measure the time from the first INVITE to the beginning of the dial tone at destination phone. In a system without CAC this time can be defined as:

$$T_{noCAC} = T_{UL} + T_{DL} + T_{pnoCAC} \quad (2)$$

In a CAC system, considering negligible the propagation time of the local network, the delay is defined as:

$$T_{CAC} = T_{UL} + T_{DL} + T_{pCAC} \quad (3)$$

Table 5 shows measured processing times added by the presence of a CAC system in the scenario.

Table 5. Processing time in ms

T_{pnoCAC}		T_{pCAC}	
mean	std. dev.	mean	std. dev.
2.36	0.52	7.87	0.83

From (2) and (3), we can obtain the delay that the CAC system introduces as:

$$T_{CAC} - T_{noCAC} = T_{pCAC} - T_{pnoCAC} \quad (4)$$

It can be shown that the CAC system introduces an additional mean delay of 5.5 ms. This time is significantly lower compared to the usual propagation and transmission Internet delays, thus the CAC system may not entail a quality impairment for VoIP communications.

The previous establishment call delays have also been compared to the ones obtained when a call is redirected to another central office, according to the CAC system decision (Table 6). The local agent acts as a redirect server forwarding the call to the PBX, and from there to another central office.

Table 6. Redirect delay in ms

T_{Pre}	
mean	std. dev.
12.84	0.72

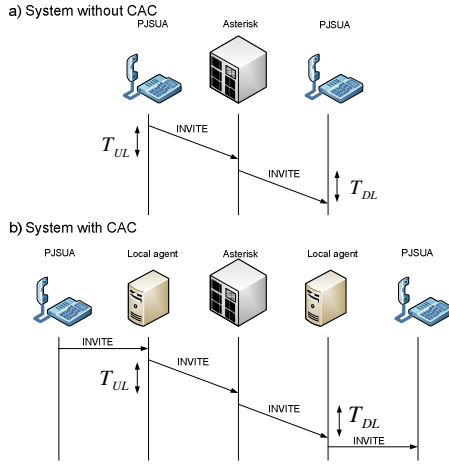


Fig. 6. Call request with and without CAC

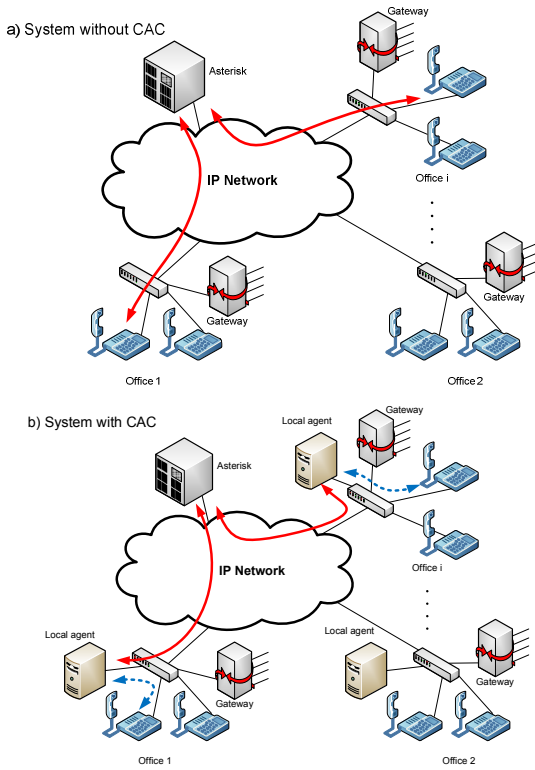


Fig. 7. Comparative of the system with and without CAC

Redirecting a call entails a total processing delay designed as T_{Pre} . We show that this delay is about 5 ms, that is, the difference between the Table 6 delays and the T_{pCAC} measured in Table 5.

Besides this additional delay, we must consider the transmission delay that brings up for the call signaling messages in each one of the links between central offices involved in the redirecting process to the PBX. This delay depends on the considered scenario and the number of redirections that a certain call needs. Let T_{UL_i} be the uplink delay of the i office, and T_{DL_i} the downlink delay. The delay of a call that has been redirected N times is defined as:

$$T_{redir} = T_{pCAC} + N \cdot T_{Pre} + T_{UL_1} + T_{DL_{N+2}} + \sum_{i=2}^{N+1} (T_{DL_i} + T_{UL_i}) \quad (5)$$

Depending on each office's delay and the number of redirections carried out, resulting T_{redir} value might be inadmissible. Thus, it is absolutely unavoidable to obtain this T_{redir} measurement for each tested environment to ensure a suitable and proper behaviour of the CAC system.

7 Conclusions

In this article, we have presented a proof of concept of a CAC system for SIP-based IP Telephony platforms, based on QoS measurements. According to these measurements, the number of lines available at the gateway and the accounting tariffs, accepts, rejects or redirects the call towards another central office. The tables on which CAC decisions are based, have been also discussed.

We have implemented a test platform composed of three central offices, based on a virtualization environment. Suitable software, PBX, SIP proxy, soft phone and gateways complete the emulated platform.

Finally, the system has been properly validated by carrying out processing delay measurements. It has been observed that these delays do not significantly impair the call establishment. Nevertheless, the call establishment delay could be highly impaired in case the number of redirections increases too much.

Acknowledgements

This work has been partially financed by RUBENS (*Rethinking the Use of Broadband access for Experience-optimized Networks and Services*) project, of EUREKA CELTIC (code EU-3187 CP5-020) European project, and the project TSI-020400-2008-020 of AVANZA I+D sub-programme, of the Spanish Ministry of Industry, Tourism and Commerce.

References

1. Bearden, M., Denby, L., Karacali, B., Meloche, J., Stott, D.T.: Assessing Network Readiness for IP Telephony. In: Proc. of the 2002 IEEE International Conference on Communications, ICC 2002 (2002)

2. One-way transmission time (recommendation g.114). International Telecommunication Union (ITU) (February 1996)
3. Chen, X., Wang, C., Xuan, D., Li, Z., Min, Y., Zhao, W.: Survey on QoS Management of VoIP. In: Proc. of the 2003 International Conference on Computer Networks and Mobile Computing. IEEE Computer Society, Los Alamitos (2003)
4. Jiang, Y., Ernstad, P.J., Nicola, V., Nevin, A.: Measurement-based admission control: A revisit. In: 17th Nordic Teletraffic Seminar (2004)
5. Rosenberg, J., et al.: SIP: Session initiation Protocol, RFC 3621 (2002)
6. Zave, P.: Understanding SIP through Model-Checking. In: Schulzrinne, H., State, R., Niccolini, S. (eds.) IPTComm 2008. LNCS, vol. 5310, pp. 256–279. Springer, Heidelberg (2008)
7. Braden, R., Clark, D., Shenker, S.: Integrated Services in the Internet Architecture: an Overview. RFC 1633 (1994)
8. Nichols, K., et al.: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474 (1998)
9. Blake, S., et al.: An Architecture for Differentiated Services, RFC 2475 (1998)
10. Yu, J., Al-Ajarmeh, I.: Call Admission Control and Traffic Engineering of VoIP. In: Second International Conference on Digital Telecommunications, IEEE ICDDT 2007 (2007)
11. Wang, S., Mai, Z., Xuan, D., Zhao, W.: Design and implementation of QoS-provisioning system for voice over IP. IEEE Transactions on Parallel and Distributed Systems 17(3), 276–288 (2006)
12. Mao, G., Habibi, D.: Loss Performance Analysis for Heterogeneous ON-OFF Sources with Application to Connection Admission Control. IEEE/ACM Transactions on Networking (February 2002)
13. 3GPP TS 24.228 v5.15.0, Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), Stage 3, R5 (September 2006)
14. SIP: Measurement-Based Call Admission Control for SIP,
http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftcacsip.pdf
15. VoIP Call Admission Control,
http://www.cisco.com/en/US/docs/ios/solutions_docs/voip_solutions/CAC.pdf
16. Braden, R., et al.: Resource ReSerVation Protocol (RSVP), RFC 2205 (1997)
17. Grossglauser, M., Tse, D.: A Time-Scale Decomposition Approach to Measurement-Based Admission Control. IEEE/ACM Transactions on Networking (August 2003)
18. Wei, H., Kim, K., Kashyap, A., Ganguly, S.: On Admission of VoIP Calls Over Wireless Mesh Network. In: IEEE International Conference on ICC 2006, June 2006, vol. 5, pp. 1990–1995 (2006)
19. Camarda, P., Guaragnella, C., Striccoli, D.: A New MBAC Algorithm for Video Streaming Based on Autoregressive Adaptive Filtering. In: IEEE International Conference on Multimedia and Expo., pp. 1512–1515 (2005)
20. Ivars, I.M., Karlsson, G.: PBAC: Probe-Based Admission Control. In: Smirnov, M., Crowcroft, J., Roberts, J., Boavida, F. (eds.) QoSIS 2001. LNCS, vol. 2156, pp. 97–109. Springer, Heidelberg (2001)
21. Cetinkaya, C., Knightly, E.: Egress Admission Control. In: Proc. IEEE INFOCOM 2000 (March 2000)

22. Alipour, E., Mohammadi, K.: Adaptive Admission Control for Quality of Service Guarantee in Differentiated Services Networks. *International Journal of Computer Science and Network Security*, IJCSNS 8(6) (June 2008)
23. Tran, H.T., Ziegler, T., Ricciato, F.: QoS Provisioning for VoIP Traffic by Deploying Admission Control. In: Burakowski, W., Bęben, A., Koch, B. (eds.) *Art-QoS 2003*. LNCS, vol. 2698, pp. 1084–1085. Springer, Heidelberg (2003)
24. Zhou, J., Ji, Z., Bagrodia, R.: TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications. In: *Proc. IEEE INFOCOM 2006* (2006)
25. Viruete, E., Ruiz, J., Fernández, J., Martínez, I.: Handbook of Research on Mobile Multimedia. In: Khalil Ibrahim, I. (ed.) *Mobility Support in 4G Heterogeneous Networks for Interoperable m-Health Devices*, 2nd edn. Idea Group Inc., IGI (2008) (in press)
26. Jiang, X., Xu, D.: vBET: a vm-based emulation testbed. In: *Proc. of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research (MoMeTools 2003)*, pp. 95–104. ACM Press, New York (2003)
27. Jones, M.T.: *Virtual Linux. An overview of virtualization methods, architectures, and implementations*,
<http://www-128.ibm.com/developerworks/library/l-linuxvirt/index.html>
28. Quetier, B., Neri, V., Cappello, F.: Selecting A Virtualization System For Grid/P2P Large Scale Emulation. In: *Proc. of the Workshop on Experimental Grid testbeds for the assessment of large-scale distributed applications and tools (EXPGRID 2006)*, Paris, France (2006)
29. Van Meggelen, J., Smith, J., Madsen, L.: *Asterisk, the future of telephony*, 2nd edn., Cap. 12. O'Reilly, Sebastopol (2005)

Towards Real-Time Stream Quality Prediction: Predicting Video Stream Quality from Partial Stream Information

Amy Csizmar Dalal*, Emily Kawaler, and Sam Tucker

Department of Computer Science, Carleton College
Northfield, MN, USA
{adalal, kawalere, tuckers}@carleton.edu

Abstract. While mechanisms exist to evaluate the user-perceived quality of video streamed over computer networks, there are few good mechanisms to do so in real time. In this paper, we evaluate the feasibility of predicting the stream quality of partial portions of a video stream based on either complete or incomplete information from previously rated streams. Using stream state information collected from an instrumented media player application and subjective stream quality ratings similar to the Mean Opinion Score, we determine whether a stream quality prediction algorithm utilizing dynamic time warping as a distance measure can rate partial streams with an accuracy on par with that achieved by the same predictor when rating full streams. We find that such a predictor can achieve comparable, and in some cases markedly better, accuracy over a wide range of possible partial stream portions, and that we can achieve this using portions of as little as ten seconds.

Keywords: Quality of Experience, Quality of Service (QoS), Streaming Media, Measurement, Performance, Reliability.

1 Introduction

Determining the subjective, user-perceived quality of a media stream in a scalable and quantifiable way is a difficult problem. As with all Internet-based applications, there is a complex interplay between network congestion conditions and the effect these congestion conditions have on application performance. Knowing how end users perceive the quality of audio and video streamed on-demand over computer networks, and the relationship between stream quality and network congestion, can lead to better design of streaming protocols, computer networks, and content delivery systems.

A number of studies have explored the idea of combining the ease and convenience of objective measurements with the information offered by a subjective

* This work is sponsored by grants from the Howard Hughes Medical Foundation and from Carleton College. Early versions of this work were sponsored by Hewlett-Packard Laboratories.

rating such as the MOS [1] to discern user-perceived stream quality. Some, like [2] and [3], correlate measurements on both the sender and receiver sides. Others, like [4] and [5], use the Emodel [6], an objective mechanism for assessing audio quality using transmission parameters. An alternate approach is to utilize application-layer objective metrics, taken at the client's machine through an instrumented media player application [5,7,8,9]. These approaches allow one to take measurements as close to the user as possible, in some cases without requiring the user's participation, providing a more accurate assessment of the state of the application at any given time.

In previous work [10,11,12], we demonstrate that objective data collected from an instrumented media player application can be used to *predict* subjective quality ratings with a high degree of accuracy (typically 70-90%) when input into a stream quality predictor that assigns ratings using a nearest-neighbor heuristic and dynamic time warping (DTW) as its distance measure. While our success rates are quite high, we base our predictions on complete stream data well after the stream has finished playing out. A more practical approach would be to predict subjective quality ratings in real time, as the stream is playing out. Modeling such a system on our previous work, such a predictor would be trained using objective and subjective measurements from past streams ahead of time, and apply this information to the task of predicting quality ratings for streams as they play out.

An important intermediate step in this process is to determine if this same stream quality predictor can accurately predict the user-perceived quality of a video stream using only partial information about the stream to be rated and/or the streams in the training set, and if so, if some portions are better or worse than others in terms of accuracy. This is the focus of this paper.

The input to our stream quality predictor consists of set of video stream state information, namely packet retransmissions, collected from an instrumented media player application for 228 video streams, with corresponding user-perceived quality ratings for these same streams. We first train the predictor using all of the available data and ratings for all streams, then test the predictor by having it predict ratings for ten-second and fifteen-second portions of these same streams. In addition, we train the predictor on various portions of the original streams, and then test the predictor on portions of the original streams as well. We compare the predictor's accuracy in these scenarios to the predictor's accuracy when training and testing on full streams. Our results show that in most cases, our predictor is about as accurate using partial stream data as it is using full stream data. We also demonstrate that we fare slightly better when we use partial stream data for both training and testing than when we train the predictor on full streams to predict the quality of partial streams, and that this holds for many combinations of training and test stream portions, even ones that are dissimilar in time from each other. In some cases, we can consistently achieve hit rates above 90%, in particular when we select similar portions (in time) from similar streams. Finally, we show that we need as little as ten seconds of data to achieve these hit rates.

The rest of this paper is structured as follows. We review the characteristics of video streams that can be exploited to infer stream quality in Sect. 2. We discuss our stream quality prediction algorithm, describe how partial stream prediction can be used to prove the feasibility of real-time stream prediction, and present the methodology we use to form partial streams from our data, in Sect. 3. In Sect. 4, we describe the source data for the experiments and the mechanism we use for evaluating predictor accuracy. Section 5 presents the results of our experiments and discusses their implications on stream quality prediction system design. We conclude the paper in Sect. 6 and highlight areas for future work.

2 Video Stream Characteristics

We have developed an instrumented version of Windows Media Player [12] that collects application-layer data about the state of a media stream at predefined intervals (currently, one second) using ActiveX hooks. Figure 1 shows an example of the data collected by our tool for a stream several minutes in duration that experiences a moderate level of network congestion. The plot illustrates a few ways in which the media player reacts to the presence of congestion on the network: for example, the number of retransmitted packets increases and the rate at which packets are received decreases as soon as congestion is detected on the network, while the number of lost packets rises later on in the plot. The plot also shows a transient period, several seconds in duration, at the start of

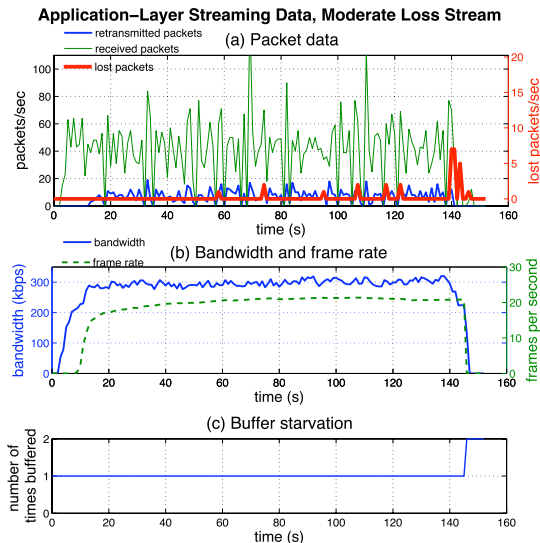


Fig. 1. Time-series data collected by the instrumented media player application. (a) Packet-level data: retransmitted packets and received packets are on the left y-axis, lost packets on the right y-axis. (b) Bandwidth and frame rate, on the left y-axis and right y-axis, respectively. (c) Buffer count, including the initial startup buffering period.

the stream, where the packet reception rate, bandwidth, and frame rate all rise to their steady-state levels as the player and server negotiate the connection between them. There is a comparable transient period at the end of the stream, where we see an uptick in the number of lost packets reported and a buffer starvation event occurring, as the player and server account for packets that will not be able to be recovered by the end of the stream.

Streams that have been exposed to similar levels of network congestion will most likely show similar patterns of retransmitted packets, lost packets, etc., even if the exact occurrences and durations do not match exactly. Streams that exhibit these similar characteristics will also exhibit similar user quality ratings, particularly once individual user biases have been accounted for. Ideally, there will be one or more measurements that most strongly reflect these quality ratings. In [12], we found that retransmitted packets are the most strongly influential on user-perceived stream quality. Thus, we can reduce the stream state information to just this one measurement over time, and discern stream similarity based on this measurement.

3 A Methodology for Real-Time Stream Prediction Using Partial Streams

Exploiting objectively-measured stream data to predict user-perceived stream quality ratings resembles problems that are classic data mining problems. By comparing patterns within the application layer metrics to user quality ratings for that stream, we can understand the effects of network congestion on user perception of stream quality.

Our stream quality prediction algorithm is described in detail in [10]; here, we briefly summarize its operation. Our particular problem calls for using knowledge of pre-labeled data to predict labels on new data [13,14,15]. The goal is to produce a predictor by training, i.e. running a data mining algorithm, on a set of labeled data. The predictor can then be tested on unlabeled data. Our data consists of a set of measurements collected from the instrumented media player on a given set of streams. The labels in this case are the quality ratings assigned by users who watched these streams as measurements were being collected (see Section 4 for details on how this data was obtained).

Our predictor uses a *nearest neighbor* algorithm, which locates all of the rated streams in the training set which are closest to the unrated stream, subject to some distance metric. A single rating is produced from the set of ratings for the closest points: if there is one nearest neighbor, assign the unrated stream that stream's rating; otherwise, compute the mean of the ratings and assign that value to the unrated stream. The distance metric used by our predictor is an extension to *dynamic time warping* (DTW), a generalization of Euclidean distance designed for use with time series data, that facilitates its use on multi-dimensional time series [16]. Briefly, DTW is based on the assumption that two time series may be quite similar, even if the precise timing between the two series is misaligned. While DTW aligns the start and end points of each time series

(stream), it allows points in mid-stream to align with the closest appropriate point. This fluidity often results in more accurate predictions and pattern identifications. A stream of unknown quality that exhibits packet loss on a periodic basis, for example, is expected to have similar quality to another stream that also loses packets periodically. However, it should not be a requirement for similarity between such streams that the packet losses occur *at precisely identical times*. To reduce the computational time and (quadratic) complexity inherent in DTW, we apply two optimizations: the popular Sakoe-Chiba band [17][18], which limits the distance that one time series can shift relative to the other; and Keogh minimum bounds [17], to quickly determine candidates for the set of nearest neighbors.

Preparing this predictor is a two step process. The first step, *training*, consists of reading in and storing the state information collected for a single stream rated by a single individual. The second step, *tuning*, consists of selecting the proper predictor parameters or inputs: K , the number of neighbors to use for predicting the quality of a stream, and w , the width of the Sakoe-Chiba bands, which we do using a leave-one-out cross-validation procedure on each training set.

In previous work, we have trained and tested this predictor using all of the data collected from a single media player application for a single user who watched and rated a particular media stream subjected to a particular level of network congestion. While doing so gives us a good idea of the accuracy of the predictor under the best of circumstances, it is not realistic. A production stream quality prediction system will have to assign ratings to incomplete streams. To mimic these circumstances, and as an important intermediate step to determine the feasibility of predicting stream quality in real time, we consider mechanisms for reducing the available information about a stream in the predictor’s training phase and test phases.

One approach is to train our predictor using all of the information available from each stream, then have the predictor assign ratings to smaller portions of the available test streams. The advantage of this approach is that the predictor does not require full stream information before assigning a rating to a test stream. The disadvantage is that DTW can perform poorly when training and test stream sizes are severely mismatched; since DTW fixes the start and end points of the streams, it compacts the longer (training) streams to match the shorter streams (to be rated), which may mean that we lose valuable information about the longer stream in the process. Another approach is to train our predictor using only the information that it is likely to have about the streams it will be rating: in this case, smaller portions of the available training streams. The advantage to this approach is that the stream sizes are similar, allowing for potentially better matches by DTW, as we have shown previously [10].

It is desirable to determine the smallest portion of a stream for which our predictor achieves accurate stream quality predictions, as well as the “optimal” location in the training and test streams from which to take these samples. A unique challenge in this case is to determine how to best select comparable intervals from the full streams, which have different durations, such that we can

easily compare shorter and longer streams. We select arbitrarily small portion sizes, ten and fifteen seconds, and divide each stream into smaller substreams of these lengths. We also divide each stream into the same number of substreams, regardless of the total length of the stream, by taking our portions at certain percentages from the start of the stream (in this case, between 1% and 90%). Thus, a thirty-second stream and a four-minute stream will yield the same number of substreams. This means that the smaller streams will be somewhat over-represented in our training set and that the longer streams will be somewhat underrepresented in our training set. However, it also means that we can easily match up substreams from different source streams, without worrying about not having an analogous period from the source stream.

Different streams will have different stream state characteristics which will vary over the lifetime of the stream. The transient and steady-state periods, for instance, will have different characteristics; the steady-state period's state information may also reflect the current level of action in the video, or the duration from the start of the stream. To determine how best to match up different portions of the stream during the training and test phases of our prediction algorithm, we use the following approach. We first train our predictor as usual with the full stream information. When testing, we rate each substream using the full stream training information. This demonstrates how well the predictor does when it has less information in the testing phase than in the training phase. We then train the predictor using, in turn, each possible substream, and then test it on each possible substream. This demonstrates how well training and test stream intervals match up when taken from similar and dissimilar points in the stream, as well as from similar and dissimilar source streams.

4 Experiments

Our data collection mechanism, testbed network, and experimental setup are described in detail in [10]; we summarize these briefly below.

Our data collection testbed consists of a set of 14 client machines on a subnet of a small campus network, and a media server on an isolated subnet with a router which runs NIST Net software [19]. The media server is a 2.4 GHz Pentium processor machine with 512 MB of RAM, running Windows Server 2003 and Windows Media Server 2003 software, streaming RTP over UDP. The NIST Net router is a 700 MHz processor machine with 512 MB of RAM, running Linux kernel 2.4.21-27 and NIST Net version 2.0.12. The client machines have 3.4 GHz Pentium processors and 1 GB of RAM and run Windows XP SP2 and Windows Media Player version 10.

Table 1 lists the source streams used in this study, which were selected to provide some variety in duration, style, content, and amount of action. NIST Net applies randomly-distributed packet losses on the testbed network, over the duration of each stream, at percentages of 0, 5, 15, and 25; there was no additional delay or delay jitter applied to the network. The network packet losses, which we determined experimentally, are higher than those typically seen

Table 1. Description of the source streams used in this study

Name	Time (mm:ss)	Action Level	BW (kbps)
Ad	0:30	Moderate	273
Trailer	2:22	High	273
News	4:09	Moderate	331

in computer networks, both to overcome the mechanisms that Windows Media Player uses to mitigate the effects of network congestion [9] and to affect the media experience in an obvious fashion that influences the streams in the same manner each time.

We showed our study participants each of the three streams twice, once with no packet loss introduced and once with either 5, 15, or 25% packet loss, blindly randomized over the participants. The participants rated the audio, video, and overall quality of each stream using seven-point scales, which allows for slightly finer granularity in participant responses [20,21]. The measurement tool collected data from each stream simultaneously. From these experiments, we collected data from a total of 38 participants and their respective client machines, yielding data for 228 streams in total.

Normalizing user ratings mitigates the factors that affect user ratings, such as individual sensitivity to encoding differences, by basing ratings on the biases of the particular user in question. We use a z-score to normalize ratings, $z_s = \frac{r_s - \bar{r}}{\sigma_r}$, where r_s is the user’s quality rating for stream s , \bar{r} is the average of the user’s quality ratings on all streams viewed, and σ_r is the standard deviation of the user’s quality ratings on all streams viewed.

We measure prediction accuracy by a *hit rate* metric, where hit rate is the percentage of time a prediction falls within 0.8 standard deviations of the user’s z-score for that stream. This corresponds to approximately plus or minus one point on the raw seven-point scale.

Using the data we collected, we first trained our predictor on each of the three full streams, then used this training data to assign ratings to the ten and fifteen second long substreams described in Section 3. We then trained the predictor on each substream and used this training data to assign ratings to each substream. In the discussion below, we refer to substreams as “partial streams”.

5 Results

In this section, we present our results for the two experiments described above: training on full streams and rating partial streams, and both training on and rating partial streams. For space reasons we only present the results for the ten-second stream portions; the results for the fifteen-second stream portions are nearly identical. As a point of reference, Table 2 lists the hit rates achieved by our predictor when both training and testing on full streams. With one exception,

Table 2. Hit rates for the stream quality predictor when training and testing on full streams

Training Stream	Test Stream			Params {K, w}
	Ad	Trailer	News	
Ad	88.2	80.3	80.3	3, 1
Trailer	72.4	89.5	80.3	6, 0
News	64.5	75.0	86.8	8, 2

the hit rates achieved by this predictor are all above 72%, with hit rates above 80% for the majority of the train/test stream scenarios.

5.1 Assigning Ratings to Partial Streams with Full Stream Training Sets

Figure 2 illustrates the accuracy of the predictor when training on full streams and assigning ratings to ten second portions of the streams. The x-axes indicate the percentage offset from the start of the stream from which the portion was taken. The plots show a clear transient period at the start of each testing stream, lasting anywhere from 7% to 30% from the start of the stream, during which hit rates are below 60%. They also show a transient period at the end of the stream for Ad, but not for Trailer or News. During the steady-state period, hit rates fluctuate between 70% and 85% when either Ad or Trailer is used as the training stream. These hit rates are comparable to slightly lower than the hit rates achieved by the predictor when training and testing on full streams. Hit rates are also comparable when the predictor assigns ratings to steady-state portions of the Ad stream when News is the training stream.

When News, the longest stream, is the training stream, hit rates decrease significantly (to below 70%) when the predictor rates portions of either Trailer or News. Here we have hit upon a possible limitation of our system: the longer the training stream, the less its characteristics match portions of the test streams. Our results indicate that this limit is somewhere between the durations of Trailer and News (2:20 and 4:10).

5.2 Assigning Ratings to Partial Streams with Partial Stream Training Sets

Figure 3 plots the accuracy of the predictor when training on and rating ten second portions of the streams. The percentages along the x- and y-axes indicate the percentage offset from the start of the stream from which the ten second portion was taken. The z-axis shows the hit rate for the predictor for a given training and test stream combination.

When Ad is the test stream (the stream to be rated), shown in the first column of plots in the figure, the best hit rates occur from about 40% of the way through the test stream until about 70% of the way through the stream, with hit rates typically between 80 and 90%. This is a significant improvement

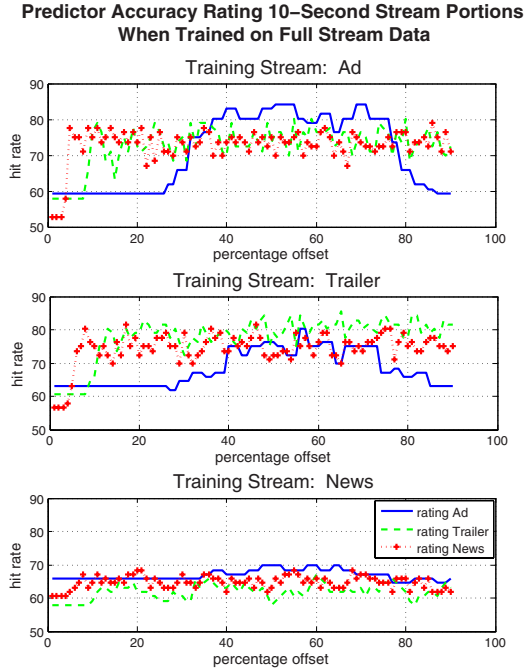


Fig. 2. Hit rates when training on full streams and testing on 10-second portions of the streams, for all combinations of training and test streams

over the predictor’s accuracy when using the full streams for training and testing (which are 72 and 65% for Trailer and News as training streams, respectively). Here, using smaller portions of streams that are dissimilar in length when both training and testing actually benefits the predictor, removing the pathologies that make it difficult to accurately match up the two streams when using DTW.

When portions of Ad are used as the training stream and portions of either Trailer or News are used as the test streams, hit rates are between 75 and 85% during the steady-state period, These hit rates are comparable to slightly lower than the hit rates achieved when the predictor trains and rates full streams.

The plots for the training/test stream combinations Trailer/Trailer, Trailer/News, News/Trailer, and News/News all exhibit similar characteristics to each other. During the steady-state periods, the predictor successfully rates the test stream between 80 and 90% of the time for News/News and Trailer/Trailer, and between 75 and 85% for News/Trailer and Trailer/News, which is comparable to the hit rates when training and testing on full streams.

Figure 4 shows the portions of the training and test streams for which the predictor is most accurate. The plots show that when the training and test stream portions are taken from the same source stream, the best hit rates cluster around the diagonal, which means that the predictor does best when the training and

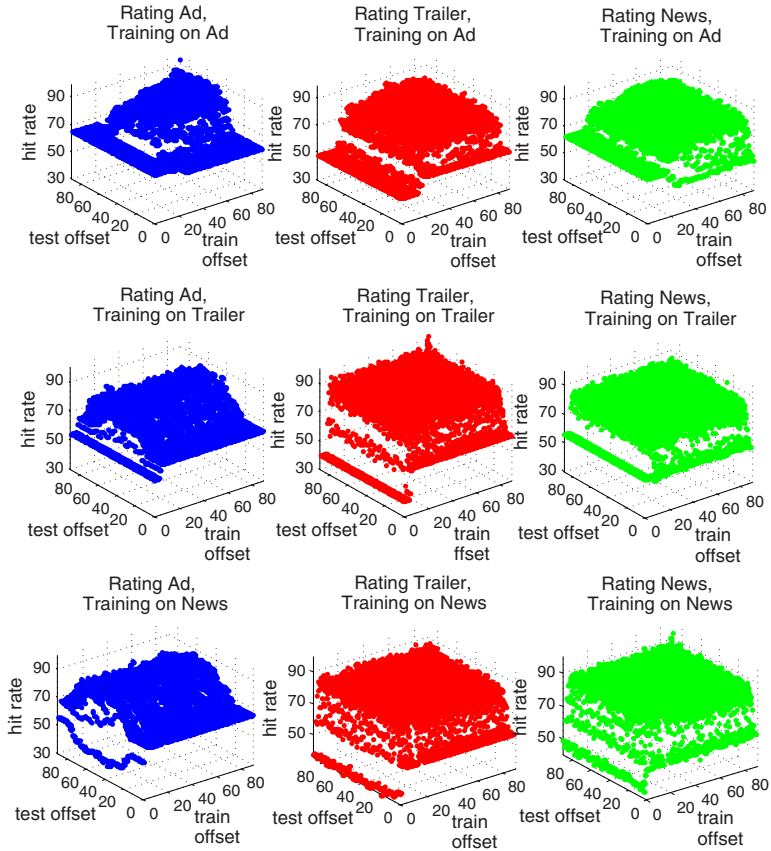


Fig. 3. Hit rates when training and testing on 10-second portions of the streams, for all combinations of training and test streams

test streams are selected not just from the same source stream, but from the same portion of the stream. In fact, the most significant result here is that the predictor is actually able to achieve hit rates over 90% for Ad and News and over 95% for Trailer under these circumstances. When the training and test streams are taken from different source streams, the best hit rates are between 85 and 90%, which is still rather high. We also see that we do not necessarily have to pull our training and test stream portions from similar time periods in the stream to achieve such high hit rates.

5.3 Discussion

Our results show that accurate predictions are quite possible, even with intervals as small as ten seconds long, when only partial information about a stream is available. In general, hit rates were at least in the neighborhood of, if not better

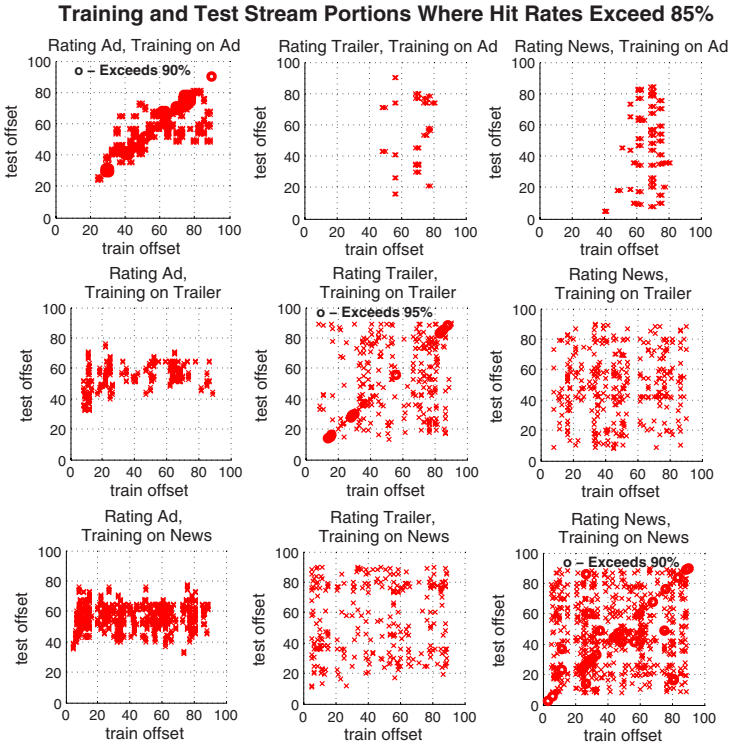


Fig. 4. Training and test stream combinations that yield hit rates above 85%. The Ad/Ad and News/News plots show data for hit rates above 88%, while the Trailer/Trailer plot shows data for hit rates above 90%. The circles indicate hit rates that are better than 90% (or 95%, in the case of Trailer/Trailer).

than, the hit rates achieved by the same predictor when using all available stream information for training and testing.

In general, a real-time stream quality prediction system should avoid training or testing during the transient period of the streams. Training and testing during these periods leads to unreliable results and inaccurate ratings, because the characteristics of this portion of the stream are dissimilar to the characteristics of the steady-state portions of the streams. With very short streams, where the transient period is relatively long compared to the length of the stream, we should also avoid training and testing during the end-of-the-stream transient period, since for short streams we often see a big uptick in certain measurements to help make up for the lack of recovery time during stream play-out. For longer streams, our results do not show an appreciably noticeable end-of-stream transient period.

If we use all available stream information (full streams) in the training phase of our predictor, then the predictor can accurately predict stream quality ratings on ten-second stream intervals between 70 and 85% of the time, assuming that

one of the shorter streams (Ad or Trailer) is used as the training stream. If we use News, our longest stream, as the training stream, then we are not able to achieve such accurate results, due to the compacting of the longer stream by DTW.

If we use partial stream information in both the training and testing phases of our predictor, then as long as the predictor avoids training or testing during the stream's transient period, it has a lot of freedom in terms of choosing appropriate training and test intervals. This is particularly true when the training and/or test stream portions are pulled from Trailer or News. This means that we can achieve results that are just as accurate, on balance, if we take the training and test intervals from very different portions of the stream as if we took them from similar portions of the stream. In a real-time stream prediction system where storage space and time to locate and load training results may be at a premium, this means we can pre-select a few portions of a stream and use any of them as our training data when assigning a rating to a new stream.

It is possible to find training and test stream portions for which the predictor can achieve better than 85% accuracy. If we can guarantee that our training and test streams have similar characteristics, as is the case when our training and test streams are both taken from Ad, or from Trailer, or from News, and take our training and test stream portions from approximately the same time period, our predictor can actually achieve hit rates above 90% (or above 95% in one case).

6 Conclusion

This paper examines the feasibility of real-time stream quality prediction, by studying whether a nearest-neighbor stream quality predictor using DTW as a distance measure can accurately rate streams based on partial stream state information. To answer this question, we examine two scenarios. In the first scenario, we train our predictor with full stream state information and attempt to rate streams where we have removed all but a small portion of stream state information. In the second scenario, we train our predictor on small portions of the full streams and then attempt to assign ratings to small portions of the full streams. We have shown that there is a wide range of training and test stream combinations that yield acceptably high hit rates, on par with or better than that achieved by the same predictor when using full stream information for both training and testing, and that we can do so using as little as ten seconds of information from each stream. This means that a stream quality prediction system operating in real time does not have to worry about using training and test streams from the same time period in the stream; training portions pulled from the end of a stream can accurately rate portions from earlier in the stream, and vice versa. We have also demonstrated that our predictor is especially accurate when we take ten second portions of the streams that are nearly identical in stream characteristics and in where in the stream they occur, achieving hit rates above 90 or even 95%. This indicates that it is possible to design a highly

accurate stream quality predictor with minimal stream information (as little as ten seconds from each stream), if we know some characteristics of the training and test streams *a priori*.

This work represents a proof-of-concept of the feasibility of real-time stream quality prediction systems, and as such there are extensions of this work that we are currently pursuing. Our data set consists of videos streamed over UDP, rather than the more ubiquitous TCP; we are currently in the process of collecting more data for streams over TCP. From a systems perspective, we are also working on a very basic prototype system to determine how best to collect, store, and evaluate stream data in real time. Finally, we are modifying the measurement tool and measurement infrastructure to enable us to collect stream ratings at intermediate points during a stream, to further improve the accuracy of our stream quality predictor.

References

1. P.910, I.T.R.: Subjective video quality assessment methods for multimedia applications. Recommendations of the ITU, Telecommunications Sector
2. Wolf, S., Pinson, M.H.: Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system. In: Proceedings of SPIE International Symposium on Voice, Video, and Data Communications, Boston, MA (September 1999)
3. Ashmawi, W., Guerin, R., Wolf, S., Pinson, M.H.: On the impact of policing and rate guarantees in Diff-Serv networks: A video streaming application perspective. In: Proceedings of SIGCOMM 2001, San Diego, CA (August 2001)
4. Clark, A.D.: Modeling the effects of burst packet loss and recency on subjective voice quality. In: Proceedings of the IP Telephony Workshop, New York (March 2001)
5. Calyam, P., Mandrawa, W., Sridharan, M., Khan, A., Schopis, P.: H.323 Beacon: An H.323 application related end-to-end performance troubleshooting tool. In: Proceedings of NeTS 2004, Portland, OR (October 2004)
6. G.107, I.T.R.: The Emodel, a computational model for use in transmission planning. Recommendations of the ITU, Telecommunications Sector (1998)
7. Wang, Y., Claypool, M., Zuo, Z.: An empirical study of RealVideo performance across the Internet. In: Proceedings of IMW 2001, San Francisco, CA (November 2001)
8. Loguinov, D., Radha, H.: Measurement study of low-bitrate Internet video streaming. In: Proceedings of IMW 2001, San Francisco, CA (November 2001)
9. Nichols, J., Claypool, M., Kinicki, R., Li, M.: Measurement of the congestion responsiveness of Windows streaming media. In: Proceedings of NOSSDAV, Kinsdale, Ireland (June 2004)
10. Csizmar Dalal, A., Musicant, D.R., Olson, J., McMenamy, B., Benzaid, S., Kazez, B., Bolan, E.: Predicting user-perceived quality ratings from streaming media data. In: Proceedings of ICC 2007, Glasgow, Scotland (June 2007)
11. Csizmar Dalal, A., Olson, J.: Feature selection for prediction of user-perceived streaming media quality. In: Proceedings of SPECTS 2007, San Diego, CA (July 2007)

12. Csizmar Dalal, A.: User-perceived quality assessment of streaming media using reduced feature sets. Technical report, Carleton College (April 2009)
13. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge (2001)
14. Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice Hall, Englewood Cliffs (2002)
15. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
16. Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., Keogh, E.: Indexing multi-dimensional time-series with support for multiple distance measures. In: KDD 2003, pp. 216–225. ACM Press, New York (2003)
17. Keogh, E., Ratanamahatana, C.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7(3), 358–386 (2005)
18. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics Speech Signal Process 26, 43–49 (1978)
19. Carson, M., Santay, D.: NIST Net: a Linux-based network emulation tool. SIGCOMM Comput. Commun. Rev. 33(3), 111–126 (2003)
20. Krosnick, J.A., Fabrigar, L.R.: Designing rating scales for effective measurement in surveys. In: Survey Measurement and Process Quality, pp. 141–165. Wiley-Interscience, Hoboken (1997)
21. Tang, R., William, M., Shaw, J., Vevea, J.L.: Towards the identification of the optimal number of relevance categories. Journal of the American Society for Information Science 50(3), 254–264 (1999)

Risk-Aware QoP/QoS Optimization for Multimedia Applications in Wireless Networks

Yanping Xiao¹, Chuang Lin¹, Yixin Jiang¹, Xiaowen Chu², and Shengling Wang¹

¹ Department of Computer Science and Technology, Tsinghua University, Beijing, China
{ypxiao, clin, yxjiang, slwang}@csnet1.cs.tsinghua.edu.cn

² Department of Computer Science, Hong Kong Baptist University, Hong Kong, P.R. China
chxw@comp.hkbu.edu.hk

Abstract. The unique characteristics of wireless networks pose a number of nontrivial challenges to multimedia applications with security and rigorous QoS requirements. Lack of adequate security protection is incapable of meeting security requirements of applications, whereas enabling excessive security services inevitably leads to further degradation in QoS due to additional computation and payload encapsulation. Early work, e.g. LAP, achieves balance by adjusting security policy according to QoS metrics; but none of them are security guaranteed. In this paper, we present an efficient risk-aware QoP (Quality of Protection) and QoS optimization algorithm for multimedia applications in wireless networks. It can achieve an optimization for QoP/QoS performance metrics through offering hierarchical security services and QoS support. Experiment demonstrates that even in high risk environments our scheme can efficiently balance QoP and QoS requirements.

Keywords: QoP, QoS, Optimization, Security, Risk-aware.

1 Introduction

With the rapid proliferation of wireless networks and real time multimedia applications, providing Quality of Service (QoS) and security protection simultaneously in an efficient manner has become a hot topic of current research in wireless networks. Real-time multimedia applications such as VoIP [2-3] and VOD have their specific QoS and security requirements. For example, VoIP applications in civilian use expect stringent delay and packet loss rate but does not expect too much on security aspect; while in military wireless networks, a majority of voice, image, video and data are required to transmit in real time and security mode, i.e., they have very stringent QoS and security requirements. Therefore, how to provide QoS and security guarantee simultaneously to meet security and performance requirements for different applications is a challenging problem. A novel mechanism is required to consider QoS and security together in a uniform and efficient way.

On the other side, in wireless networks, the bandwidth of a link is unpredictable and possibly very low, and the channel capacities and error rates are time-varying, which makes it harder to design multimedia application with stringent QoS requirements than in wireline networks. Furthermore, the shared channel in wireless

networks makes it easy for data intercepting and tempering and leads to the breach of security; however, enabling excessive security services inevitably leads to further degradation in QoS due to additional computation and payload encapsulation especially in wireless networks with stringent resource constraints. So how to make a tradeoff between security and QoS is a critical issue for multimedia applications with rigorous security and QoS requirements in wireless networks.

Compared with wireline networks, there are more challenges in security and QoS assurance in wireless networks, such as (1) Highly dynamic topology demand to negotiate and configure QoP/QoS policy; (2) The shared media and attenuation of channel make it more difficult in QoS assurance; (3) Adopting individual QoP configuration in heterogeneous devices can not satisfy the performance requirement.

The experiments demonstrate that implementing stronger security services can affect QoS seriously such as packet loss and delay [15] in wireless networks. Most of the existing schemes focus on guaranteeing either security or QoS, but not both. With the prevalence of wireless multimedia applications, some schemes [11-15] integrating security with QoS have been proposed, which improve the grade of security service as much as possible in the precondition of satisfying the QoS requirements. Once QoS decreases or the systems dissatisfy the QoS requirement of applications, they improve QoS metrics by adopting weaker security services.

However, there are still some shortcomings in such schemes: (1) Some applications do not require QoP such as common web applications, and some do not need over-high QoP. Over-high QoP policies not only lead to further degradation in QoS performance, but also decrease utilization rate of system resource. (2) The method of adjusting QoP according to QoS metrics is vulnerable to attack and information leak on account of adopting low security service to improve QoS. (3) Directly adjusting QoP policies only according to the QoS metrics cannot efficiently guarantee the satisfaction of QoS. It should extensively consider other optimization mechanisms such as traffic classification [1], channel access [2], and packet scheduling [3], etc.

For real time multimedia applications with stringent QoS and security requirement, the existing schemes are neither security guaranteed nor QoS guaranteed. To provide QoS and security support simultaneously perfectly, we propose a risk-aware QoP/QoS model for wireless multimedia applications, which can efficiently select appropriate QoP policy to avoid the impact of excessive QoP on QoS by apperceiving the status of security and system resource. Moreover, it can guarantee QoS by dynamically adjusting QoS policies, and thus makes a nice tradeoff between QoP and QoS.

In summary, the contributions of this paper mainly include three aspects: (1) It provides a novel, adaptive and risk-aware multi-level QoP/QoS optimization model to achieve a balance between QoP and QoS. (2) It provides a multi-level QoS model which can be used in real time QoS assurance. (3) It progresses toward a notion of QoP in security comparable to the notion of QoS in networking.

The rest of the paper is organized as follows. The related work is given in section 2. Section 3 respectively introduces multi-level QoP model, multi-level QoS model, and risk-aware multi-level QoP/QoS model in detail. A generic multi-level QoP/QoS framework is presented in section 4. The optimization algorithms between QoP and QoS are discussed in section 5, followed by the experiments in section 6. Finally, section 7 gives conclusions along with future works.

2 Related Work

Compared with QoS, the concept of QoP has surfaced in the literatures for the latest several years. The main idea of QoP is to provide multi-level security services for different users and traffic and to meet the requirements in increasingly complicated environments, and has been focused especially in wireless networks.

Ong et al. [5] firstly presents a QoP framework which provides differential security service levels for mobile multimedia applications with heterogeneous devices in wireless networks. Based on idea of [4], Agarwal et al [9] extend the QoP model and study the impact of different security policy on QoS in wireless LAN networks. Furthermore, since authentication is the first line of defense to provide security service, Liang et al. [5-8] deeply study the impact of challenge/response authentication on QoS performance in wireless LANs. To decrease the impact of authentication on performance, Schneck et al. [10] propose a dynamic authentication protocol to improve the performance of the system. Although the above schemes considered and studied the impact of security on QoS, they don't consider how to achieve the optimization between QoP and QoS.

It is no doubt that providing differentiable security service can decrease security impact on performance, but it is not enough to provide QoS assurance. Therefore, some schemes integrated QoP with QoS have been presented. He et al. [11-12] proposes an integrated solution to delay and security support in wireless networks aiming to wireless applications with stringent delay and security requirement. Almost at the same time, Agarwal et al. [14, 15] develops a link-aware protection (LAP) mechanism to coordinate security and QoS in wireless networks. However in MANET there is little research integrated QoS and security as well, So Shen et al. [13] presents a security and QoS self-optimization mechanism to achieve the optimization. However, the schemes are not security-guaranteed, and the policy of improving QoS through adjusting QoP is not enough to provide QoS assurance.

3 Risk-Aware Mutli-level QoP and QoS Model

In this section, an integrated risk-aware multi-level QoP / QoS optimization model is proposed based on the multi-level QoP model and the multi-level QoS model.

3.1 Multi-level QoP Model

Definitions 1: QoP is defined as the protection quality of security services by using security metrics such as authentication, confidentiality, integrity, non-reputation and availability et al. and formally described as a quintuple vector $P = \langle Au, C, I, N, A \rangle$, where Au, C, I, N and A denotes authentication, confidentiality, integrity, non-reputation and availability respectively. $Au \in [0, 1]$, where 0 denotes no authentication service is provided and 1 denotes the highest authentication mechanism. Its quantification can be referred to implementation mechanisms, strength of algorithms, or length of key. And the same definition applies to C, I and N . A is a real number between 0 and 1, and denotes the probability of whether the service is available.

We can select different security metrics to embody the levels of security services. Without loss of generality, assuming that the available security policy includes m security features such as authentication and confidentiality, and every security feature includes n optional configuration. Therefore, security policy can be described as an $m \times n$ matrix, and each element P_{ij} in the matrix denotes one policy configuration of security feature i .

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{m1} & \cdots & P_{mn} \end{bmatrix}$$

Different security policy achieves different protection levels. To describe the protection levels of different security policy, we define a protection levels function g on matrix P as

$$G = g(P) \quad (1)$$

Similarly, matrix G has a form like matrix P , every element in G has one-to-one mapping to every element in P . The protection levels of security policy with the same feature can be compared directly. The composite protection levels of security policy involving multi security features need introduce the definition of QoS Composite Metric (QCM).

Definition 2: QCM is a real number which combines multi security metric and reflects the quality of multi security features; its definition is as follows.

$$qop = \sum_{i=1}^m P_i \times \omega_i \quad (2)$$

where P_i denotes protection levels of security feature i , $i \in [1, m]$, ω_i denotes the weight of security feature i , which satisfies

$$\sum_{i=1}^m \omega_i = 1. \quad (3)$$

Security features can not avoid influencing the performance of system. To describe it, we introduce the definition of Performance Impact Matrix (PIM).

Definition 3: PIM reflects the impact of security policy matrix P on QoS. For any element P_{ij} in P , we can introduce a function f to denote its performance impact, which is defined as

$$C_{ij} = f(P_{ij}). \quad (4)$$

C_{ij} denotes the impact of security policy P_{ij} on QoS, $i \in [1, m]$, $j \in [1, n]$, it corresponds to a vector $C_{ij} = [d_{ij}, j_{ij}, b_{ij}, l_{ij}]$, where d_{ij} , j_{ij} , b_{ij} and l_{ij} denotes delay, jitter,

$$C = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \vdots & \vdots \\ C_{m1} & \cdots & C_{mn} \end{bmatrix}$$

bandwidth, and packet loss rate, respectively. Therefore, we can get PIM C , which can be denoted as matrix P .

Different applications require different quality of security services, even the same application may require different quality of security services. Multi-level QoP model can provide tunable security services according to different security requirements, especially in wireless networks with stringent resource constraints, which can be described as $qop_i \in \{qop_1, qop_2, \dots, qop_n\}$, where qop_i denotes individual security feature or multi security features, i denotes the quality level of security service.

Generally, the impact of QoP on QoS is positive correlation to the level of QoP. When the performance is decreasing, we can improve QoS by decreasing QoP.

Lemma 1: For a given multi-level QoP model with $qop_i \in \{qop_1, qop_2, \dots, qop_n\}$, the higher qop_i is, the more the impact on system performance is.

Intuitively speaking, QoP levels of security services are positively correlated with time complexity and space complexity of cryptographic algorithms. Whereas the higher the complexity is, the bigger the impact on performance is.

3.2 Multi-level QoS Model

Different applications have special QoS requirements. For example, for VoIP, its QoS metrics may be denoted as, $\{delay < 150 \text{ ms}, Jitter < 50\text{ms}, Bandwidth > 64\text{kbs}, lose \text{ rate} < 3\%\}$.

Most of QoS requirements can be described as a quadruple $\zeta = \langle d, j, b, l \rangle$, where d, j, b, l respectively denotes delay, jitter, bandwidth, and packet loss rate. Each QoS parameters can be divided into n levels, which can be described as a matrix S ,

$$S = \begin{bmatrix} d_1 & d_2 & \dots & d_n \\ j_1 & j_2 & \dots & j_n \\ b_1 & b_2 & \dots & b_n \\ l_1 & l_2 & \dots & l_n \end{bmatrix},$$

Each element in matrix S indicates a range of a QoS parameter. For delay, jitter, and packet loss rate, the lower the level is, the higher the quality is. For bandwidth, the higher the level is, the higher the quality is.

Individual QoS parameter can be compared directly; the comparison of multi-QoS parameter is required to introduce the definition of Satisfied Degree of QoS (SDQ) and Composite Satisfied Degree of QoS (CSDQ).

Definition 4: SDQ is defined as a real number and is denoted the degree of QoS satisfaction of applications.

Taking delay as an example, and assuming that $delay < D \text{ ms}$, the delay of epoch t is d , then the satisfied degree of delay is calculated as follows,

$$qos_d(d) = \begin{cases} \frac{D-d}{D} & d \in [0, D] \\ 0 & d > D \end{cases} \quad (5)$$

Similarly, the satisfied degree of Jitter, Bandwidth and Loss rate can be respectively define as follows.

$$qos_j(j) = \begin{cases} \frac{J-j}{J} & j \in [0, J] \\ 0 & j > J \end{cases} \quad (6)$$

$$qos_b(b) = \begin{cases} 0 & b < B \\ \frac{b}{B} & b \geq B \end{cases} \quad (7)$$

$$qos_l(l) = \begin{cases} \frac{L-l}{L} & l \in [0, L] \\ 0 & l > L \end{cases} \quad (8)$$

Definition 5: CSDQ is defined as follows,

$$qos(d, j, b, l) = qos_d(d) \times \omega_d + qos_j(j) \times \omega_j + qos_b(b) \times \omega_b + qos_l(l) \times \omega_l \quad (9)$$

where $\omega_d, \omega_j, \omega_b$ and ω_l denote the weight of each parameters respectively, and $d \in [0, D], j \in [0, J], b \in [B, +R], l \in [0, L]$.

The value of $\omega_d, \omega_j, \omega_b$ and ω_l in Eq. (9) is relative to certain types of network traffic. For example, the most concern are delay and packet loss rate for VoIP traffic, so both of the corresponding weight may be set to 0.4, and the other may be set to 0.1 equally.

QoS parameters fluctuate constantly in real scenarios. Multi-level QoS model can be described as $qos_i \in \{qos_1, qos_2, \dots, qos_n\}$, where qos_i indicates a single QoS parameter or a composite QoS parameter, i denotes the level of QoS.

If the system fails to guarantee qos_i , we can relax from qos_i to qos_{i-1} . If the system cannot meet the lowest level qos_1 , we can only drop the application.

3.3 Risk-Aware Multi-level QoP/QoS Model

Risk-aware multi-Level QoP/QoS model can efficiently decrease the impact of QoP on QoS by selecting multi-level QoP according to the risk level of system, and can also provide multi-level QoS service by adjusting QoS policies according to the monitored QoS metrics, which is especially appropriate to applications with stringent QoS and security requirement in wireless networks.

Assuming that $S_0 = [d_0, j_0, b_0, l_0]$ denotes QoS of the application without introducing any security services, $f(P_j)$ denotes the impact of security policy P_j on QoS, so the objective is maximum QoS in the precondition of security assurance, that

is to find a m dimensional vector Γ , which can maximize QoS of applications with m security policy configurations,

$$\Gamma = \arg \max_{j_i \in [1, n]} (\alpha \times qos(d_0, j_0, b_0, l_0) - \beta \times qos(\sum_{i=1}^m f(P_{ij}))) \quad (10)$$

Subject to

$$\begin{cases} g(P_{ij}) \geq R_i, i \in [1, m]; \\ d_0 \in [0, D]; \\ j_0 \in [0, J]; \\ b_0 \in [B, +R]; \\ l_0 \in [0, L]; \end{cases} \quad (11)$$

where α, β denotes a constant respectively, they depend on the status of wireless network, and often are set to 1. In Eq. (11) $g(P_{ij})$ denotes the protection level of security service, R_i denotes the risk levels to be introduced in the following.

Theorem 1: For a risk-aware multi-level QoP/QoS model, if it satisfies QoS/QoP requirements of an application, the following conditions $qos(d_0, j_0, b_0, l_0) + qos(\sum_{i=1}^m f(P_{ij})) > 1$ must hold.

Proof: For an application with rigorous QoS requirement, it must satisfy conditions $d_0 < D, j_0 < J, b_0 \geq B, l_0 < L$. To provide normal service to an application, its QoP levels must larger or equal the risk levels. However QoP cannot avoid affecting QoS,

$$qos(\sum_{i=1}^m f(P_{ij})) = qos(\sum_{i=1}^m C_{ij}) \quad (12)$$

Extending Eq. (12), we can get the following equation.

$$\begin{cases} qos(\sum_{i=1}^m f(P_{ij}))_d = qos(\sum_{i=1}^m d_{ij}) = \sum_{i=1}^m \frac{D-d_{ij}}{D} \\ qos(\sum_{i=1}^m f(P_{ij}))_j = qos(\sum_{i=1}^m j_{ij}) = \sum_{i=1}^m \frac{J-j_{ij}}{J} \\ qos(\sum_{i=1}^m f(P_{ij}))_b = qos(\sum_{i=1}^m b_{ij}) = \sum_{i=1}^m \frac{b_{ij}}{B} \\ qos(\sum_{i=1}^m f(P_{ij}))_l = qos(\sum_{i=1}^m l_{ij}) = \sum_{i=1}^m \frac{L-l_{ij}}{L} \end{cases}$$

We take delay as an example. If the in-equation $qos(d_0) + \sum_{i=1}^m qos(d_{ij}) \leq 1$ holds, then

$\sum_{i=1}^m d_{ij} + d_0 \geq D$ must holds. Obviously it contradicts total $d < D$, so the equation holds. For the other parameters, the conclusion also holds. \square

Improving S_0 or decreasing QoP levels can improve QoS, S_0 can also be improved by QoS mechanisms such as resource reservation, access control, packet scheduling and traffic classification etc. Decreasing QoP levels can be achieved through altering

security policy with higher levels to those policy with lower levels. Therefore, it is required to introduce the notion of risk levels.

Definition 6: Risk levels correspond to the protection levels of security features and can be described as a column vector $[r_1, \dots, r_i, \dots, r_m]$, $r_i \in [1, n]$.

Wireless networks are subject to many attacks such as data intercepting, tempering etc; therefore, there exist many potential risks. The degree of risks is relative to the environments and application requirements. Risk levels justly reflect the extent of potential threats. We assume IDS/IPS can report the potential risk levels in real time, and then we can adjust security policy according to the risk levels.

Normal services can be provided only when the QoP levels are higher than risk levels. So there is a problem how to select security policy. The related algorithm can be described as follows:

Algorithm 1. Optional-Policy-Matrix-Cal

```

Optional - Policy - Matrix - Cal( $P, D$ ) {
01:   $D \leftarrow P$ 
02:  for  $i = 1$  to  $m$  get( $r_i$ )
03:  for  $i = 1$  to  $m$ 
04:    for  $j = 1$  to  $n$ 
05:      if  $P[i, j] < r_i$  then
06:         $P[i, j] = null$ 
07:  }
```

Supposing that the risk levels vector from IDS/IPS is $R = [1, 2, \dots, n]$, then we can get an optional policy matrix D ,

$$D = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ 0 & P_{22} & \dots & P_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & P_{mn} \end{bmatrix}$$

Every element in matrix D indicates an optional security policy configuration. The total of schemes provided by the system is $\prod_{i=1}^m N_i$, where N_i denote the number of optional policy.

To understand the impact of QoP on QoS further, we get a theorem as follows.

Theorem 2: For a risk-aware multi-level QoP and QoS model, if there exists a security policy p which satisfy QoS and QoP requirements simultaneously, then its impact on QoS must satisfy the equation, $\sum_{i=1}^m C_{i_r} \leq f(p) \leq \sum_{i=1}^m C_{i_n}$, where $r_i \in [1, n]$ denotes the current risk levels of the corresponding security feature.

Proof: If the security policy p satisfies requirement of QoS/QoP simultaneously, then the protection levels corresponding to the security policy p are not smaller than risk

levels vector $r_i, i \in [1, m]$ at least. So if we can choose the security policy with the same level of risk, the impact I on QoS is calculated as below.

$$I_{min} = \sum_{i=1}^m f(P_{ir_i}) = \sum_{i=1}^m C_{ir_i} \tag{13}$$

When choosing the security policy with the maximum QoP level, the impact of QoP on QoS is calculated as

$$I_{max} = \sum_{i=1}^m C_{in} \tag{14}$$

When choosing the security policy in the optional sets at random mode, the impact of QoP on QoS is calculated as

$$I = \sum_{i=1}^m \frac{1}{n-r+1} \sum_{j=r}^n C_{ij} \tag{15}$$

According to Lemma 1, $f(p)$ must be limited in the range of value of the Eq. (14) and Eq. (15), so the equation is easy to be proofed.

4 Generic Risk-Aware Multi-level QoP/QoS Framework

In this section we present a generic multi-level QoP/QoS framework for wireless multimedia networks. The progress of wireless technique enables more mobile devices such as PDA, Mobile Phone to access internet through wireless networks. Deploying real time multimedia applications in these devices not only require considering their capability of CPU, Memory and IO fully, but also require providing enough security. So we provide a generic risk-aware multi-level QoP/QoS framework which can provide tunable security service and performance support, especially appropriate to wireless networks with stringent resource constraints.

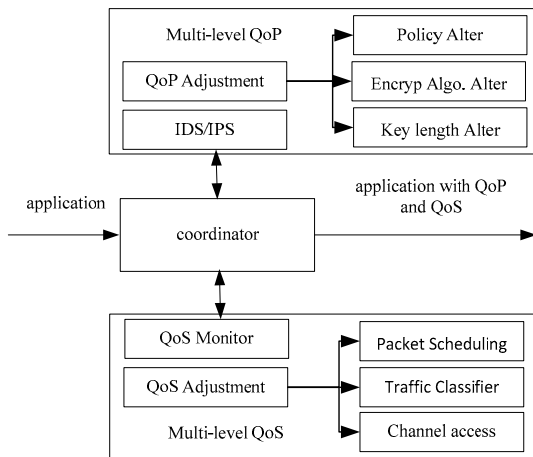


Fig. 1. A Generic Tunable QoP and QoS Framework

Fig. 1 shows a generic multi-level QoP/QoS framework, which consists of multi-level QoP module, multi-level QoS module, and a coordinator module. The coordinator module is charge of negotiating and providing service for QoP/QoS requirements. When the negotiation is agreed, it constantly adjusts the QoP levels and QoS policy to meet the requirement of an application according to system risk levels and status of system resource.

Multi-level QoP module includes IDS/IPS and QoP adjustment components. IDS/IPS monitors the incoming and outgoing network traffic, the system files and the running process, and reports the risk according to the abnormality distance [19]. Multi-level QoP components provide the response according to the risk levels issued by the IDS/IPS. The most likely responses [20] in wireless network is to reinitialize communication channels between nodes, the corresponding measures may include the adjustment of the security policy such as adding the authentication, integrity, or confidentiality mechanisms, altering encryption algorithms, or altering key lengths of algorithms. In this paper we focus on the alteration of cryptographic algorithm and key lengths.

Multi-level QoS module consists of QoS monitor and QoS adjustment components, they coordinate to satisfy QoS requirement of applications. QoS monitor module is uninterruptedly monitoring the QoS metrics of real-time applications. The metrics of performance monitor includes delay, jitter, and bandwidth, and packet loss rate. If the metric is in the critical rang of its value, QoS adjustment components will be invoked to improve QoS performance according to the status of system resource. QoS adjustment components mainly consist of traffic classifier, packet scheduling, and channel access. In this paper, we only focus on the packet scheduling through altering the priority of packets.

5 Risk-Aware Multi-level QoP/QoS Optimizations Algorithms

Since the implementation of QoP inevitably affects QoS, to decrease the impact of QoP on QoS and meet QoS requirements simultaneously in wireless networks with limited resources, some effective algorithms are required to achieve the optimization and tradeoff between QoP and QoS. The objective of optimization is to maximize QoS under the condition of certain QoP.

To accomplish the optimization and tradeoff between QoP and QoS in wireless networks, we adopt the method of dynamically selecting security policy, which is especially fit to the devices with limited computation resource. When risk levels change, we select security policy with the same QoP level. In such a way, we can decrease the impact of QoP on QoS to minimum. At the same time, if QoS can't be satisfied because of variability of channel in wireless networks, we can also call QoS adjustment components such as altering the priority of packets to improve QoS according to QoS monitor.

Our algorithms consist of two parts, one is risk-aware multi-level QoP adjustment algorithms, and the other is QoS optimization algorithms. Some parameters are listed in Table 1.

Table 1. Parameters in Algorithms

Parameters	Descriptions
qos^*	A variable denoting an initial QoS value being negotiated by the parties, satisfying $qos^* \in \{qos_1 \leq qos_2 \leq \dots \leq qos_{max}\}$
qop^*	A variable denoting an initial QoP value being negotiated by the parties, satisfying $qop^* \in \{qop_1 \leq qop_2 \leq \dots \leq qop_{max}\}$
pr^*	A variable denoting an initial priority value being negotiated by the parties, satisfying $pr^* \in \{qop_1 \leq qop_2 \leq \dots \leq qop_{max}\}$
$S(t)$	QoS value of an application in real time
$R(t)$	Risk level issued by IDS in real time
$pr(t)$	Priority of packets for an application in real time

In Table 1, qos^* and qop^* are calculated by the Eq. (9) and Eq. (2), respectively, pr^* denotes the priority of packets. qos_{max} and qos_1 denote the highest QoS level and the lowest QoS level respectively. qop_{max} and qop_1 denote the highest and lowest level of security services. pr_{max} and pr_1 denote the highest and lowest priority level.

5.1 Risk-Aware Multi-level QoP Adjustment Algorithms

Wireless networks are more subject to attacks, and thereby some strong security mechanisms are adopted. However, these mechanisms may affect the performance severely. To decrease the impact to minimum, risk-aware multi-level QoP adjustment algorithm is presented in this paper. When the risk levels are larger than the level of security protection, QoP adjustment components will be enabled to improve QoP level in order to weaken potential threats. There are many modes to adjust security policy. The minimization protection mode may be the best choice. QoP adjustment algorithm is described as follows.

Algorithm 2. QoP-Adjust

```

QoP_Adjust( $qop^*$ ){
01: if  $R(t) > qop^*$  and  $R(t) = qop_i$  then
02:    $qop^* = qop_i$  and Call  $QoS\_Ada(qos^*)$  and  $Pr\_Adjust(pr^*)$ 
03: else
04:   keep  $qop^*$  unchanged
05: }
```

5.2 QoS Optimization Algorithms

The shared media and attenuation of channel in wireless network make QoS assurance more challenging than in wireline networks, therefore more QoS mechanisms should be considered when providing QoS support. Once performance

decreasing, the mechanisms can be invoked. We take priority adjustment as an example to introduce QoS optimization algorithms. We assign a priority of packets according to the QoS requirements in initialization, and then adjust the priority of packets to improve QoS according to the monitoring of QoS in run-time. At the same time, the system adaptively changes QoS according to the priority and the monitored QoS metrics. The optimization algorithms consist of QoS adaptive algorithm and priority adjustment algorithm, which are depicted as below respectively.

When the priority of packets is maximal and detectable QoS level is larger than minimum level, qos^* can be degraded properly. But it should be ensured in the pre-specified range. When qos^* is smaller than the minimum and qop^* level is larger than the risk level, qop^* level can be decreased to improve qos^* , otherwise the application should be discarded.

Algorithm 3. QoS-Ada

```

QoS_Ada( $qos^*$ ){
01: if  $pr(t) = pr_{max}$  and  $S(t) < qos^* \neq qos_1$  then  $qos^* = qos_{i-1}$ 
02: if  $pr(t) = pr_{max}$  and  $S(t) < qos^* = qos_1$  then
03:     if  $R(t) < qop^*$  then  $qop^* = qop_{i-1}$  until  $qop^* = R(t)$ 
04:     else Drop the application
04: if  $pr(t) < pr_{max}$  and  $qos_{i+1} > S(t) > qos_i$  then  $qos^* = qos_i$ 
05:     else  $qos^* = qos_{i+1}$  until  $qos^* = qos_{max}$ 
06: }

```

When detected qos^* is smaller than pre-specified requirement, we can increase the priority of packets to improve qos^* . When qos^* is larger than pre-specified requirement, we can decrease the priority the algorithms is described as below.

Algorithm 4. Pr-Adjust

```

Pr_Adjust( $pr^*$ ){
01: if  $S(t) < qos^*$  and  $pr^* = pr_i$  then
02:      $pr^* = pr_{i+1}$  until  $pr^* = pr_{max}$ 
03: else if  $S(t) > qos^*$  then
04:      $pr^* = pr_{i-1}$  until  $pr^* = pr_1$ 
05:     else
06:     keep  $pr^*$  unchanged
07: }

```

Although we can adjust the priority of packets to improve qos^* , it cannot assure the satisfaction of QoS. Our algorithms consist of monitoring of QoS performance, so we can adjust pr^* and QoS policy to assure QoS before exacerbation of QoS.

6 Experimental Studies

VoIP application is a very typical of multimedia application. In this section we take wireless VoIP as an example to simulate and demonstrate our model and algorithms.

6.1 Experiment Setup

We set up a wireless LAN test bed to simulate 802.11 wireless network transmit. The test bed consists of two servers (an ftp server and a voice gateway server) behind an access point, and three mobile clients. The access point and stations send packets by the rate 11Mbps. We assume use of G.729 as audio codec, and emulate VoIP traffic by UDP packets with 32 byte data at the rate of 50 packets percent second between a voice gateway and a voice station. We also simulate the background data stream by ftp uploading and downloading between one ftp server and two data stations. All the computers are running RedHat Linux 9 with kernel version 2.6.9-5. Fig. 2 shows the topology of the test bed, in the figure the voice station send packets to voice gateway, Data station 1 and 2 uploads and downloads files to simulate the background streams respectively.

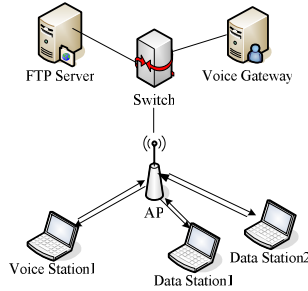


Fig. 2. Network Topology for Experiment

6.2 Experimental Results

In this subsection, we take DES, AES with three different key lengths as encryption policy and MD5, SHA1 as integrity policy to simulate the impact of different QoP policies on QoS, and we also assume risk levels with one-to-one mapping relation to QoP policy. Then we simulate the Risk-aware Multi-level QoP and QoS algorithms (RMQQ) as described in Section 5.

To measure the impact of QoP on QoS in wireless VoIP applications, we used a VoIP performance metric R proposed in [16-17], which takes into account delay, loss rate, and the type of the encoder. R is defined in Eq. (16), which reflects QoS of VoIP application and should provide a value above 70. If the value is blow 70, the quality of VoIP can't meet the requirements.

$$R = 94.2 - 0.024d - 0.11(d - 177.3)H(d - 177.3) - 11 - 40 \log(1 + 10e) \quad (16)$$

where d denotes delay, it consists of codec delay, playback delay and network delay. If we consider the delay caused by different security level services, the delay should include qop delay denoted by d_{qop} .

$$d = d_{code} + d_{playout} + d_{network} + d_{qop} \quad (17)$$

Some experimental parameters and the combination of algorithms are shown in Table 2 and Table 3 respectively.

Table 2 lists some delay parameters in Eq. (18). $d_{network}$ and d_{qop} are parameters which are attained by our experiment.

Table 2. Experimental Parameters

Parameters	Value
d_{code}	25ms
$d_{playout}$	60ms
$e_{playout}$	0.005
sending rate	50packet/sec
packet size	32byte
wireless bandwidth	11Mbps

Table 3 lists the combination of encryption algorithms and integrity algorithms. DES and AES can achieve encryption feature. DES is replaced by AES in wireless networks by virtue of lower security level of DES in comparison with AES. Both MD5 and SHA1 can achieve integrity, the security level of MD5 is lower than SHA1, but its efficiency is more than SHA1.

In order to calculate QoP levels of different algorithms combinations in Table 3, we compile Cypto Libraries [18] in gcc at the voice station IBM T60 with the configuration Genuine intel® CPU T2400@1.83GHz and 512M memory. We do 1000 experiments to get the average throughput of algorithms as shown in Table 4.

Table 3. QoP Level of Different Policy Configuration

QoP Policy	QoP Level	System Risk Level
DES-MD5	1	1
AES128-MD5	2	2
AES192-MD5	3	3
AES256-MD5	4	4
DES-SHA1	5	5
AES128-SHA1	6	6
AES192-SHA1	7	7
AES256-SHA1	8	8

Table 4. Throughput of Different Algorithms (Mbps)

Alg.	Key Length(bits)	IBM T60
DES	56	109.2Mbps
AES	128	85.0Mbps
AES	192	79.5Mbps
AES	256	72.9Mbps
MD5	-	288.7Mbps
SHA1	-	103.8Mbps

According to the average throughput of each algorithm, we can easily to sort and set their QoP levels due to the positive correlation between the QoP levels of algorithms and compute complexity.

In order to simulate RMQQ algorithm, assume that we can get the risk levels from IDS or IPS. Because the risk levels are varied with the potential attacks and the environments, so in the experiment we adopt a random method to simulate the variation of risk levels. We assume that risk levels change once per hour, and we don't distinguish specific security features and adopt composite QoP level in Eq. (2) to correspond to risk levels. When risk levels are varied, we adopt QoP policy with the same level or the higher level. System risk levels and QoP policy corresponding to risk levels are also shown in Table 3.

We adopt the security algorithms in Table 4 to simulate the adjustment of QoP policy, i.e, we encrypt packets in information source and decrypt them in receiving end. At the same time, data station 1 uploads big movie files to the ftp server and data station 2 downloads files from the same ftp server. Our monitoring time is about 1 minute, 3000 packets. We do experiments about one hour for every algorithm combination, then we average the delay and packet loss, the results is shown in Fig. 3 and Fig. 4, respectively.

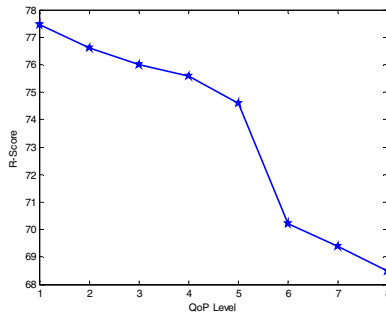


Fig. 3. Variation of R-Score with QoP Level

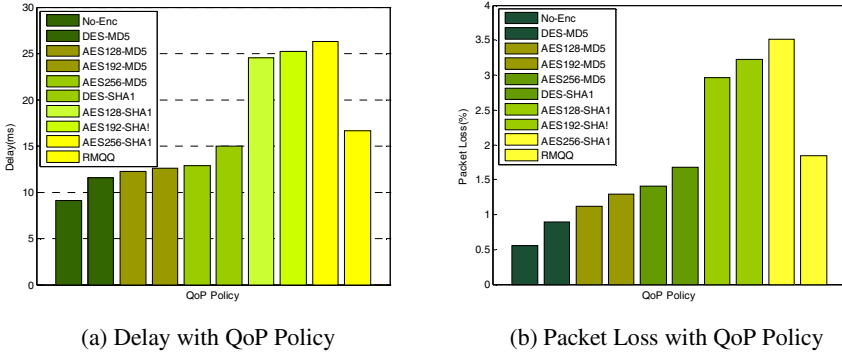


Fig. 4. Performance of QoP Policy

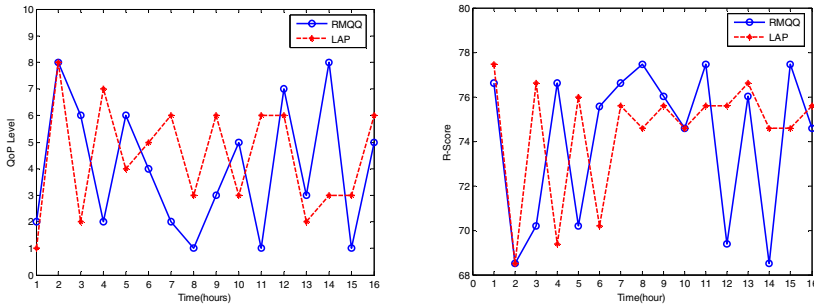


Fig. 5. Variation of Risk level for RMQQ/LAP Fig. 6. Variation of R-Score for RMQQ / LAP

Fig. 3 shows the relation between QoP level and R-Score. From the graph we can draw a conclusion that R-Score decreases with QoP level increasing.

Fig. 4 shows the VoIP delay and packet loss rate with different QoP policy. The higher the QoP level of security policy is, the more the performance impact on QoS is. RMQQ gets the result in the range of the minimum and maximum owing to adjusting the policy according to the variation of risk levels.

In comparison with LAP [15], we simulate the snapshot of RMQQ and LAP. Fig.5 shows the variation of QoP policy with the risk levels in RMQQ and LAP. We choose QoP policy in minimum protection mode because of the negative correlations between QoP and QoS. The line of RMQQ accords with the variation of risk level, so the line of the risk level variation is not depicted in the Fig. 5 and Fig. 6. As for LAP, there is a probability of near 50% that QoP levels are lower than risk levels. That is because LAP adjusts QoP policy according to QoS. Relative to RMQQ, LAP cannot guarantee QoP level is higher than risk level, so it suffers from more security threats.

Fig. 6 shows the impact of RMQQ on QoS, and the impact of LAP on QoS in the threshold of packet loss 1.8% and threshold of delay 13ms. The average R-Score of two schemes are 74.11 and 74.44 respectively and are very close. But when the risk levels are higher, R-Score of LAP is a little better than that of RMQQ, that is because that LAP does not consider the potential risk, and change QoP policy according to

QoS only. When the risk is lower, RMQQ choose the QoS policy with lower level while the QoS policies of LAP varies little, and their R-Scores are 76.0048 and 69.39 respectively. Relative to LAP, VoIP quality of RMQQ increases about 10%. From Fig.5 and 6, the fluctuation of scheme RMQQ seems rapider than LAP, that is because we assume that the variation of risk levels occurs every an hour in interval at random. However in reality the changing of risk levels often varies slowly.

7 Conclusion

In this paper, a risk-aware multi-level QoS/QoS optimization model is presented, which can efficiently solve real time multimedia applications with security and stringent QoS requirement in wireless networks. It can dynamically adjust QoS policy according to the risk issued by IDS, and provide multi-level security services, decrease the impact of QoS on the QoS, and guarantee QoS of applications. Experiments demonstrate that the risk-aware multi-level QoS/QoS model can not only provide multi-level QoS/QoS services, but also achieve optimization and make a nice tradeoff between QoS and QoS. The more important is that it extends the early idea of adjusting QoS policy according to QoS metrics only and integrates security with QoS into one model, which can efficiently coordinate QoS with QoS. Future work will introduce heuristic algorithm to solve the multi-level QoS/QoS optimization problem.

Acknowledgments. This research was supported by the National Grand Fundamental Research 973 Program of China (No.2010CB328105) and the National Natural Science Foundation of China (No.60970101, No. 60673187, No. 60872055 and No. 60803123).

References

1. Barry, M.G., Campbell, A.T., Veres, A.: Distributed Control Algorithm for Service Differentiation in Wireless Packet Network. In: Proc. IEEE INFOCOM (2001)
2. Hanley, G., Murphy, S., Murphy, L.: Adapting WLAN MAC Parameters to Enhance VoIP Call Capacity. In: Proc. of the 8th ACM MSWiM 2005, October 2005, pp. 250–254 (2005)
3. Yu, J., Choi, S., Lee, J.: Enhancement of VoIP over IEEE 802.11 WLAN via Dual Queue Strategy. In: Proc. IEEE ICC (2004)
4. Ong, C.S., Nahrstedt, K., Yuan, W.: Quality of Protection for Mobile Multimedia Applications. In: Proc. IEEE ICME, vol. 2, pp. 137–140 (2003)
5. Liang, W., Wang, W.: An Analytical Study on the Impact of Authentication Local Area Networks. In: Proc. IEEE 13th Intel. Conf. and Networks (ICCCN 2004), pp. 361–366 (2004)
6. Liang, W., Wang, W.: A Quantitative Study of Authentication Networks. In: Proc. IEEE INFOCOM, vol. 2, pp. 1478–1489 (2005)
7. Liang, W., Wang, W.: On Performance Analysis of Challenge/Response Based Authentication in Wireless Networks. *Computer Networks* 48(2), 267–288 (2005)
8. Wang, W., Liang, W., Agarwal, A.K.: Integration of Authentication and Mobility Management in Third Generation and WLAN Data Networks. *Wireless Comm. and Mobile Computing (WCMC)* 5(6), 665–678 (2005)

9. Agarwal, A.K., Wang, W.: On the Impact of Quality of Protection in Wireless Local Area Networks with IP Mobility. *Mobile Networks and Applications* 12, 93–101 (2007)
10. Schneck, P.A., Schwan, K.: Dynamic Authentication for High performance Networked Application. In: Proc. of the sixth International Workshop on QoS, pp. 127–136 (1998)
11. He, W., Nahrstedt, K.: An Integrated Solution to Delay and Security Support in wireless network. In: Proc. IEEE WCNC, Las Vegas, vol. 4, pp. 2211–2215 (2006)
12. He, W., Nahrstedt, K.: Impact of Upper Layer Adaptation on End-to-End Delay Management in Wireless Ad Hoc Networks. In: Proc. IEEE Real-Time Embedded Technology and Application Symposium, RTAS 2006, April 2006, pp. 59–70 (2006)
13. Shen, Z., Thomas, J.P.: Security and QoS Self-Optimization in Mobile Ad Hoc Networks. *IEEE Trans. on Mobile Computing* 7(9), 1138–1151 (2008)
14. Agarwal, A.K., Wang, W.: DSPM: Dynamic Security Policy Management for Optimizing Performance in wireless networks. In: Proc. IEEE MILCOM 2006, pp. 1–7 (2006)
15. Agarwal, A.K., Wang, W., Gupta, R.A., Chow, M.: LAP: Link-Aware Protection for Improving Performance of Loss and Delay Sensitive Applications in Wireless Lans. In: Proc. IEEE MILCOM 2007, pp. 1–7 (2007)
16. Cole, R., Rosenbluth, J.: Voice over IP performance monitoring. *ACM Comput. Commun. Rev.* 31, 9–24 (2001)
17. ITU-T Recommendation G.107, The E-model, a Computational Model for Use in Transmission Planning (1998)
18. Dai, W.: Crypto++, <http://www.eskimo.com/weidai/cryptlib.html>
19. Hariri, S., Qu, G., Modukuri, R., Chen, H., Yousif, M.: Quality-of-Protection (QoP) – An Online Monitoring and Self-Protection Mechanism. *IEEE Journal on Selected Areas in Communications* 23(10) (2005)
20. Mishra, A.: Security and Quality of Service in Ad Hoc Wireless Networks. Cambridge University Press, Cambridge (2008)

COCONET: Co-operative Cache Driven Overlay NETwork for p2p Vod Streaming

Abhishek Bhattacharya, Zhenyu Yang, and Deng Pan

Florida International University, Miami FL 33199, USA
{abhat002,yangz,pand}@cs.fiu.edu

Abstract. Peer-to-Peer (P2P) approaches are gaining increasing popularity for video streaming applications due to their potential for Internet-level scalability. P2P VoD (Video On-Demand) applications pose more technical challenges than P2P live streaming since the peers are less synchronized over time as the playing position varies widely across the total video length along with the requirement to support VCR operations such as random seek, Fast-Forward and Backward (FF/FB). We propose *COCONET* in this paper, which uses a distributed cache partly contributed by each participant thereby achieving a random content distribution pattern independent of the playing position. It also achieves an $O(1)$ search efficiency for any random seek and FF/FB operation to any video segment across the total video stream with very low maintenance cost through any streaming overlay size. Performance evaluation by simulation indicates the effectiveness of *COCONET* to support our claim.

Keywords: peer-to-peer, Video On-Demand, streaming, overlay network, co-operative cache.

1 Introduction

Today's Internet provides a powerful substrate for real-time media distribution due to the widespread proliferation of inexpensive broadband connections which makes live streaming and on-demand media applications more important and challenging. As mentioned in [1], YouTube has about 20 million views a day with a total viewing time of over 10,000 years till date which clearly makes VoD streaming to be one of the most compelling Internet applications. P2P approach of content distribution has already being proved to be useful and popular for file sharing and live streaming systems with a plethora of applications found in the Internet. P2P based design can achieve significant savings in server bandwidth for VoD systems also, as stated in [2]. But, unlike live streaming applications, very few P2P VoD systems have being implemented and successfully deployed over the Internet. In order to alleviate server load, the state-of-art P2P VoD systems allow peers exchange video blocks among each other having overlapped playing positions. In a VoD session, the users watching the same video may well be playing different parts of the stream, and may issue VCR commands at will to jump to a new playback position leading to fundamentally lower levels

of content overlap among the peers and higher need for frequently searching new supplying peers. It is observed that efficient neighbor lookup is important for supporting VCR operations. [9] presents a detailed analysis on a large scale P2P VoD system and enumerates the major design challenges. Among these, the fundamental challenge in designing a P2P VoD system lies in offering VCR operations such as random seek/FF/FB which require greater control over the coordination of peers. This is very unlike live streaming where the users are always in the same playing position and have no VCR related operations.

We propose *COCONET*, a novel and efficient way of organizing peers to form an overlay network for supporting efficient streaming and neighbor lookup for continuous playback or FF/FB VCR operations. *COCONET* utilizes a co-operative cache based technique where each peer contributes a certain amount of storage to the system in return for receiving video blocks. *COCONET* uses this co-operative cache to organize the overlay network and serve peer requests, thereby reducing the server bottleneck supporting VCR related operations. In current P2P VoD systems, peers share video segments only with nearby (forward/backward) neighbors based on its playing position [1]. We envision a problem with this type of content distribution scheme. In highly skewed viewing patterns, most of the peers are clustered around a particular playing position and very few peers are scattered at different positions throughout the video length, thereby the peers may not find any or very few neighbors to satisfy their demand. *COCONET* avoids this situation and is able to serve peers that are at random playing positions which are not at all related to the sender's playing position.

In order to find new supplier peers at different parts of the movie length, P2P VoD systems need to maintain an updated index of the live peers with their available video segments. Currently most deployed P2P VoD systems rely on centralized trackers for maintaining the index. This mechanism imposes a huge query load on the tracker with the expansion of the system. *COCONET* does not use indexing at the tracker. Instead, the tracker only maintains a small subset of live peers which is queried only once as a rendezvous point when a new peer joins the system. Each *COCONET* peer builds an index based on the co-operative cache contents which helps to find any supplier peer for any video segment throughout the entire video length.

Our main contributions in this paper are: (1) We are able to achieve a search efficiency of $O(1)$ during continuous playback or random seek to any position across the entire video stream with a high probability; (2) The control overhead of *COCONET* is also low to maintain the overlay structure upon any peer dynamics when the system is being subjected to heavy churn and moreover it has better load balancing and fault tolerance properties; and (3) One of the attractive features of *COCONET* is a distributed contributory storage caching scheme which helps to spread the query load uniformly through the overlay and organizes the overlay in a uniform and randomized fashion which makes the content distribution independent from playing position.

The remainder of this paper is organized as follows. In Section 2 we survey related work from the literature. Section 3 presents the preliminary design structure and key ideas. In Section 4 we discuss the detailed protocols in *COCONET* and analyze their performance. Section 5 presents performance evaluation of *COCONET* using simulations and we conclude in Section 6.

2 Related Work

The most studied overlay design for video streaming in the literature is tree and mesh structure. Over the last decade, many proposals have been put forward such as P2Cast [13] and oStream [11] where the basic stream is provided by an application layer multicast tree which searches for appropriate patching streams from the root peer. Similar to the tree-based schemes, P2VoD [3] organizes nodes into multi-level clusters according to their joining times. The major problem with these kind of overlays is its difficulty to maintain a consistent structure which is vulnerable in a highly dynamic environment typical of P2P based VoD systems. Multi-tree approach was proposed by SplitStream [12] where the video stream is divided into multiple sub-streams using coding techniques. One stream is sent through each tree which achieves better resilience to tolerate failures since even if one tree fails, the peers will continue to receive video blocks through the other trees with a possibly degraded quality.

PROMISE [4] uses mesh-based overlays for streaming. Although they improve bandwidth utilization by fetching data from multiple peers, supporting VCR operations in VoD service such as random seek is not easy since it is difficult to have a neighbor lookup mechanism to locate supplier nodes. Mesh-based overlays are useful for distributing the content but is not so effective for searching which is one important criterion for supporting VCR operations in a VoD system as indicated in [2].

A dynamic skip list based overlay network is proposed in [2], where all the peers are connected sequentially according to their playback progress at the base layer of the skip list and each peer may also randomly connect to a few adjacent peers on the higher layers. The lookup efficiency is shown to be $O(\log N)$ where N is the total number of peers. A ring assisted overlay is proposed in [5], where each peer maintains a set of concentric rings with different radii and places neighbors on the ring based on the similarity with their cached content with a search complexity of $O(\log(T/w))$ where T is the video size and w is the buffer size. Many DHT based approaches have also been proposed such as [6], but a DHT lookup takes logarithmic messaging complexity with respect to the number of peers in the system. With playback progress, cached blocks are frequently flushed off from the buffer which will cost a DHT update and this will incur a lot of overhead in the long run. InstantLeap [10] proposes a hierarchical overlay network which is based on the playing position of the peer. Peers are divided into a number of groups where each peer belongs to one group at a time and maintains limited membership information of all the other groups by exchanging random messages which helps to perform any random seek operation

in $O(1)$ messaging overhead. *COCONET* also achieves a $O(1)$ lookup complexity but with reduced protocol overhead than [10]. This can be observed from the fact that [10] builds its index based on each peer’s playing position i.e. available segments in the playing buffer. So whenever any peer changes its playing position due to continuous playback/leap and moves from one group to another, the index needs to be updated which involves a lot of messaging overhead. In contrast, *COCONET* is completely independent of playing position and builds its index based on the stable storage buffer which is kept unchanged as long as the peer is in the system. Thus, *COCONET* completely avoids the costly and frequent index update operation as the peers change their playing position and also helps for better segment availability of the entire system.

3 Design Principle

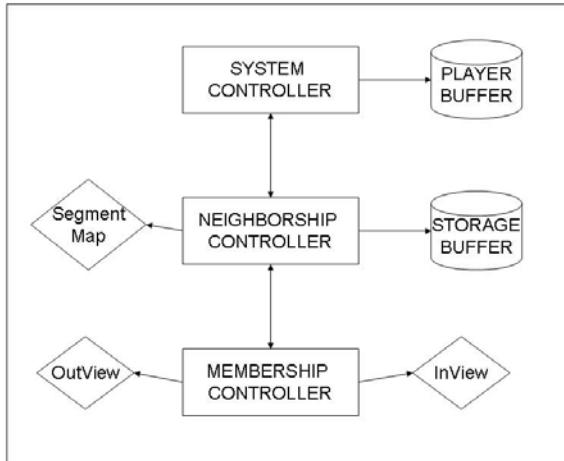
We present the basic system model in this section. We have N peers in the system and a Video Server S which stores the entire video with an upload capacity of S_u Mbps. A centralized tracker T maintains s_T , a set of t live peers in the system and is updated with a periodicity of t_T . The video is divided into M segments and the play time for each segment is t_M with a data rate of D . Each peer contributes a part of storage space to the system which is defined as the *storage buffer* (aka. *storage cache* or *co-operative cache*) with a size of b segments. For the sake of simplicity, we assume segment level granularity at all the levels of *COCONET*. Each peer also has a *playing buffer* of size k segments which contains the current playing segment and a few consecutive segments for supporting continuous playback and pre-fetch. The entire system parameters are listed in Table 1 for easy reference, with some of them explained in later sections. The system architecture diagram of a *COCONET* node is shown in Figure: 1.

3.1 Membership Management

Gossip-based algorithms have recently become popular solutions for message dissemination in large-scale P2P systems. In a typical gossip-based scheme, each peer sends a message to a set of randomly selected nodes which in turn repeat the process in the next round. The gossiping continues until the message has reached everyone. The inherent random property of gossip helps it to achieve resilience to failures and enable decentralized operations but at the cost of redundancy. The two most important knobs of gossip are: *fan-out* and *tll*. Fan-out is the number of gossip partners maintained by each peer and *tll* is the number of gossip rounds before the message is discarded. As stated in [7], for gossip to be successful the partner list should be a randomized subset of the entire population and should be refreshed before each gossip round. In *COCONET*, the membership is managed by disseminating join messages randomly to a set of peers which in turn forward to some other peers or keep it with a certain probability, thereby keeping the overlay connected. We employ this mechanism for membership management which is similar to SCAMP [7] with some modification which will be described later. Based on the join messages, each peer p maintains the *InView* (peers

Table 1. List of System Parameters

Definition	Notation
Number of Peers	N
Video Server	S
Tracker	T
Tracker Size	s_T
Number of total Video Segments	M
Play time of one video segment	t_M
Size of storage buffer	b
Size of playing buffer	k
Failure tolerance	c
TTL for Join Message	ttl_j
TTL for Gossip Message	ttl_g
SegmentMap	S_M
TimeOut for SegmentMap	t_S
Initial Buffering delay	t_b
Gossip Periodicity	t_g
Tracker Update Periodicity	t_T
Peer Upload Capacity	P_u
Peer Download Capacity	P_d
Server Upload Capacity	S_u
Data Rate	D
Storage Buffer Request Timeout	t_s
Playing Buffer Request Timeout	t_p

**Fig. 1.** System view of a *COCONET* node

which know the existence of p) and *OutView* (peers that p knows to exist) set as a partial view of the entire system which helps to facilitate node join/leave operations.

3.2 Co-operative Caching

As already mentioned, each peer in the system has to contribute a part of its storage as *storage cache* to serve other peers. Contribution awareness is already very popular in P2P file sharing applications which helps in increasing system resource and is also employed in VoD systems such as [9]. But to efficiently manage this distributed storage for better performance in alleviating server load is still a significant challenge. *COCONET* tries to solve this problem where each peer on joining the system randomly caches b segments in its buffer in the hope of serving other peers when it is required. The segments in the storage cache remain unchanged as long as the peer remains in the system. This randomly distributed cache helps to increase system stability as there is very little chance that any video segment will be unavailable in any of the participating peers. This in turn translates to server load alleviation to a large extent which is another advantage for *COCONET*. The major chance for segment unavailability in *COCONET* is due to insufficient bandwidth resources for which the peer will be forced to query the Video Server, S .

3.3 Neighborhood Management

As mentioned, efficient neighbor lookup is one of the key requirements in VoD systems for supporting VCR related operations. Each *COCONET* peer achieves this by maintaining a *SegmentMap*, S_M which is basically a list of M entries with the i -th entry, S_M^i representing the set of peers that have cached the i -th segment in their storage buffer. The underneath mechanism is a gossip-based algorithm which helps to disseminate S_M information through the entire overlay. The gossiping is done through the peer list of *OutView* which is constructed during the join operation. The gossip-based scheme will help to fill up S_M for each peer. So, essentially S_M serves as an index for the entire set of M segments and is utilized for neighbor lookup. It is trivial to observe that VCR operations can easily be satisfied by looking up S_M for the corresponding peers containing the required segments and downloading from them. Given, S_M is correctly maintained, any lookup operation can be satisfied in $O(1)$ messaging complexity by *COCONET*. We also maintain an additional failure tolerance factor c , which means that we keep c distinct peer entries for each entry, S_M^i in *SegmentMap*. Since there are a total of M segments in S_M , so the total number of entries in the *SegmentMap* of a *COCONET* peer comes to cM . This tolerance factor, c is a design choice and can be tuned according to application demand which will essentially help to tolerate peer departure/failure during churn conditions. We discuss the detailed protocol in Section 4.2.

3.4 Content Distribution Pattern

Existing P2P VoD systems distribute content from the playing buffer which is highly synchronized according to the playing position and requires any peer to download a video segment from another peer within the same playing segment or the next few consecutive segments. In highly skewed viewing patterns, if a peer issues a request for a segment with no nearby peer then the system fails to answer and has to resort for server resource. As a contrast, *COCONET* utilizes the storage buffer for content distribution and so is independent of playing position making any peer to download from a random peer with a completely different playing position. Thus, the previous situation due to highly skewed viewing patterns will have less severe impact in *COCONET* since the content distribution pattern is completely randomized.

4 Detailed Protocol

One of the important design goals of *COCONET* is to populate S_M as quickly as possible with a tolerance factor of c . This means that there should be c neighbors on average for each SegmentMapID, S_M^i . The fill-up size of S_M should be cM which is the target value for each peer. The importance of S_M in *COCONET* is obvious since all the neighbor lookup is performed through it and efficient maintenance of S_M is critical for system performance. To achieve a total size of cM , it needs to contact at least cM/b neighbors since each neighbor has b segments in storage buffer. An important theorem from SCAMP [7] states that, given a group size as N and the partial view size maintained at each peer as $O(\log N)$, the probability for a gossip to reach every member in the group converges to $e^{-e^{-c}}$ provided the link/node failure probability is not greater than $c/(c+1)$. *COCONET* exploits this theorem to reach a group size of cM/b by maintaining a partial view (i.e., OutView) size of $\log(cM/b)$. The partial view represents a randomized subset of the total number of peers in the system and thus, we use the partial view as gossip partners which will help to effectively disseminate the information among the participants. For gossip to be successful, logarithmic number of neighbors are required for information dissemination and within logarithmic rounds there is a high probability that the information reaches every member in the group, as pointed out in [7]. So, *COCONET* sets information dissemination gossip fan-out to be $\log(cM/b)$ and within $\log(cM/b)$ rounds of gossiping there is a high probability that the storage buffer information of the source peer has reached the required number of peers. Thus, there is a high probability that total list size for S_M to approach cM in logarithmic gossip rounds only.

4.1 Protocol for Join/Leave Operation

Each *COCONET* node is provided with a unique identifier. Tracker, T maintains a partial list of t live peers (t is maintained through periodic updating by the peers). Each joining peer initially contacts the tracker to acquire a *contact peer*.

Then the joining peer sends a *join* message to the contact peer. The join protocol is very similar to SCAMP [7] with a little tuning. The join message is a 3-tuple of $\langle \text{sender peer ID}, \text{join peer ID}, \text{ttl}_j \rangle$, where sender peer is the one that sends the join message, join peer is the one that have initiated the join protocol for entering the system and ttl_j refers to the number of hops by the join message before it is killed. ttl_j is set for limiting the number of join rounds so that it may not move for an infinite number of times and is generally killed whenever any peer receives the same message for more than 10 times by simply discarding the received join message. Any peer other than the contact node, receiving a join message either keeps it or forwards it to a random neighbor from its OutView with a probability proportional to $\log(cM/b)/\text{OutView.size}$. This helps to keep the OutView size close to $\log(cM/b)$ with a high probability and will be used later for gossiping to disseminate information of S_M . The pseudo-code of *join* is listed in Table: [2]

Table 2. Join Protocol

```

At Join Node:
  contact_node ← Tracker
  OutView ← OutView ∪ {contact_node}
  send_join(contact_node, join_node, _)

At Contact Node:
  InView ← InView ∪ {join_node}
  forall n ∈ OutView
    send_join(n, join_node, ttl_j)

At Other Nodes:
  if (ttl_j == 0)
    return /* drop the join msg if ttl expired */
  with probability log(cM/b)/OutView.size do
    if (join_node ∉ OutView)
      OutView ← OutView ∪ {join_node}
    else
      choose randomly n ∈ OutView
      send_join(n, join_node, ttl_j - 1)

```

The protocol for leave operation involves the modification of InView which is essentially a list consisting of nodes which contains its nodeID in their partial views. The leaving node simply informs $c + 1$ random neighbors in InView to replace its nodeID with a random neighbor selected from the partial view of the node that invoked the leave operation. Then, it informs the rest of the neighbors in InView to simply remove its nodeID entry without replacing it. This protocol is entirely local and requires no global information. The protocol is simple and we skip its pseudo-code for brevity of space.

4.2 Protocol for SegmentMap Exchange

The SegmentMap exchange is the essential component of *COCONET* which helps to populate S_M by gossiping with neighbors in OutView. To achieve reliability, gossip sends a lot of redundant messages across the communication channel which is not suitable for bandwidth hungry streaming applications. So we need to tune the gossip protocol to avoid sending excessive messages after a certain system criterion is met. The gossip messages are sent according to some probability proportional to $(\frac{c}{\text{avg. entry size of } S_M})$. This ensures that gossiping will be switched off when the average entry size of S_M comes close to c with a high probability. During peer churns, *COCONET* detects and removes the dead entries from S_M and if the average entry size of S_M goes below c , gossiping will be switched on automatically in the next cycle resulting in repopulating S_M . The gossip message is a 4-tuple $\langle \text{sender node ID, this} \rightarrow \text{node ID, ttl}_g, \text{SegmentMap information} \rangle$ where ttl_g is set to be $O(\log(cM/b))$ for effective information dissemination as discussed above. Any peer receiving a gossip message employs a push-pull based dissemination mechanism wherein the receiver peer sends its S_M information to the sender and also updates its S_M from the sender. The pseudo-code of S_M exchange is listed in Table: [3](#).

Table 3. SegmentMap Exchange Protocol

At Sender Node:

```

avg_entry_size  $\leftarrow \sum_{i=1}^M S_M[i].size / M$ 
with probability  $\propto \frac{c}{\text{avg\_entry\_size}}$  do
  choose randomly  $n \in \text{OutView}$ 
  send_gossip( $n, \text{node\_ID}, \text{ttl}_g, S_M$ )

```

At Receiver Node of gossiped S_M^s :

```

if ( $\text{ttl}_g == 0$ )
  return /* drop since ttl expired */
for  $i \leftarrow 1$  to  $M$  do
   $S_M[i] \leftarrow S_M[i] \cup S_M^s[i]$ 
  choose one random  $n \in \text{OutView}$ 
  send_gossip( $n, \text{node\_ID}, \text{ttl}_g - 1, S_M^s$ )

```

4.3 Protocol for Caching

This protocol is very simple where each *COCONET* peer randomly selects b segments for caching. After joining and the SegmentMap established, it sends download request to other peers if any entry is found in S_M . If it receives a positive reply within timeout from any peer p then it downloads from p . Otherwise, it requests the server S for the segment.

4.4 Protocol for Retrieving Segments

One of the most frequent operations in *COCONET* is the lookup operation for supporting continuous playback or random seek. Usually the query will be for a particular segment i and the system is required to return a neighbor list where each of the neighbors contain the segment i in its storage buffer. It is trivial to observe that for lookup operation to be successful in *COCONET* it is essential that S_M is maintained correctly with sufficient number of entries to tolerate failures. As mentioned, *COCONET* tries to keep c neighbor entries for each segment so that a maximum of $c - 1$ failures can be tolerated without disrupting system performance. *COCONET* maintains the overlay network ordered on the basis of storage buffer blocks and does not consider the playing buffer for message dissemination. The pseudo-code of *look-up* is listed in Table: [4](#).

Table 4. Lookup Protocol

```

Input: Query for segment  $q$ 
At Node  $n$ :
  for  $i \leftarrow 1$  to  $b$  do
    if (storage_buffer[ $i$ ] ==  $q$ )
      return storage_buffer[ $i$ ]
  multicast_download_request( $S_M[q]$ ,  $q$ )
  wait for availability reply with timeout
  if (timeout == false)
    select peer  $p$  with earliest reply timestamp
    send_download_request( $p$ ,  $q$ )
    receive segment from  $p$ 
    return
  else
    send_download_request( $S$ ,  $q$ )
    receive segment from  $S$ 

```

5 Experimental Evaluation

In this section we present our simulation results for *COCONET*. We have implemented a discrete event simulator in C++ supporting an overlay size of 10,000 or more simultaneous peers. We have used GT-ITM [\[8\]](#) to generate the underlying physical network for our simulations based on transit-stub model. The network consist of 15 transit domains, each with 25 transit nodes and a transit node is connected to 10 stub domains, each with 15 stub nodes. We randomly choose peer from the stub nodes and place the video server on a transit node. The delay along each link was selected proportional to the Euclidean distance between the peers. We have set our simulation settings to be $P_u = 1$ Mbps, $P_d = 4$ Mbps and $S_u = 80$ Mbps with $D = 500$ kbps and the total viewing length of the video to be 128 minutes. Each segment size is set to be 3.7MB which corresponds to one minute video length. Each experiment was run for a length of 7,500 seconds.

The peers join the overlay following a Poisson arrival model with arrival rate, $\lambda = 0.1$. The peer departure pattern follows an exponential distribution with an expected life time of 20 minutes. The other static parameters of our simulation are: $M = 128$, $t_j = 25$, $t_M = 60$ sec, $t = 250$, $k = 2$ segments, $b = 4, 8, 16$ segments, $c = 4$, $t_S = 5$ sec, $t_g = 20$ sec, $t_T = 50$ sec, $t_s = t_p = 25$ sec. We do not assume any transmission error in channels. For designing a VoD system it is important to efficiently utilize the upload bandwidth of all the peers in the system since it is the most scarce system resource and so we perform our discussion of experimental analysis to its usage efficiency. We avoid peer download analysis since download bandwidth is less likely to be the system bottleneck as we assumed it to be four times compared to upload bandwidth for each peer in our simulation settings. We also assume streaming a single video in our simulation scenario which is more simple to analyse.

5.1 Server Load

One of the most important objectives of P2P VoD system is to reduce server load. Figure 2 shows the server load with varying overlay sizes in *COCONET*. Server load is measured on a per streaming session basis, where it is measured by the percentage ratio of the total number of downloaded segments from the server to the total number of downloaded segments by all peers in the entire session. As we can observe from Figure 2, there is a slight increase of server load with increase in overlay size which is around 2% rise for every increase of 2,000 in overlay size. This is a very narrow increase rate and so we can conclude that *COCONET* scales well to large overlay sizes. Another interesting fact to observe is that, for similar overlay sizes, server load is greatly reduced on increasing the size of storage buffer. This can be intuitively justified by noticing that, the overall system demand remains same but system availability increases since there will be more number of available

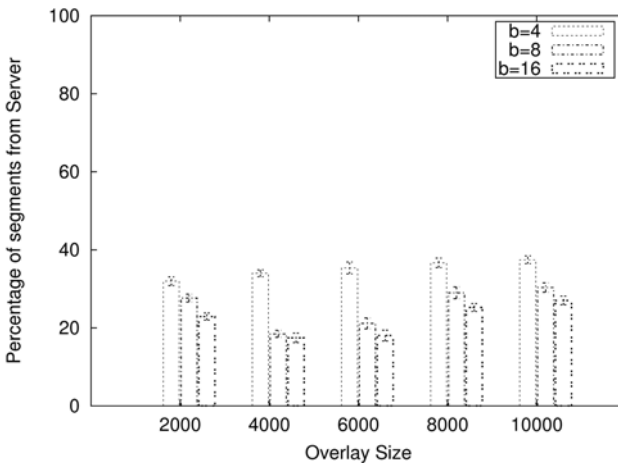


Fig. 2. Server load for varying overlay sizes

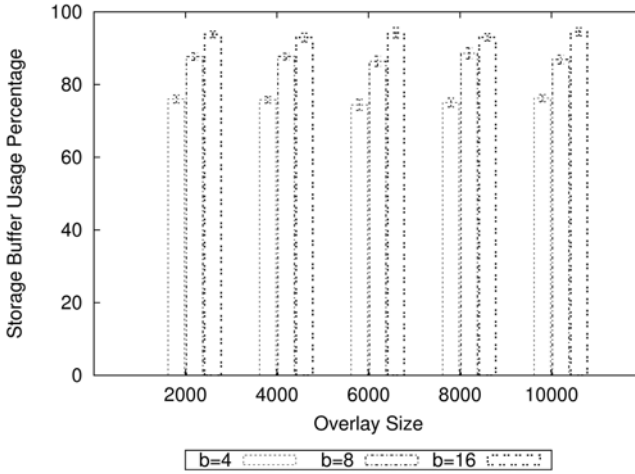


Fig. 3. Storage buffer efficiency for varying overlay sizes

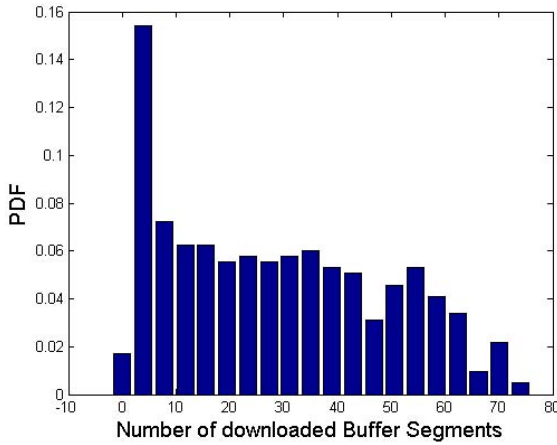


Fig. 4. Probability density plot for a function of number of downloaded buffer segments in a 10,000 node network

segments with higher value of b . Analyzing more specifically, we observe that there is a significant amount of server load alleviation when b goes up from 4 to 8 with around 15% reduction for 4,000 and 6,000. This effect is not so prominent when b goes from 8 to 16 which is around 4% in average.

5.2 Storage Buffer Usage Efficiency

In this section we study the buffer usage efficiency as this will be an important indicator for *COCONET* system performance. For each session, we measure the

total number of storage buffers available in the system which is a constant, $b \times N$ and the total number of storage buffers that have been downloaded one or more times. We calculate the buffer usage efficiency by taking the ratio of the previous two parameters and plot the results in Figure 3. With a higher value of b , there are more number of available segments in the system which in turn helps to distribute the segments more efficiently among the peers. We also plot the probability density function of storage buffer usage for one streaming session in Figure 4 with an overlay size of 10,000 for $b = 8$ and we observe that the majority usage pattern is uniform excepting a somewhat higher usage at one point.

5.3 Load Balancing

In this section we experiment on the available upload bandwidth usage for each peer. We plot the results in Figure 5 which corresponds to the aggregated bandwidth utilization efficiency for all peers per streaming session. The plots show the average efficiency calculated over all the peers for a streaming session. We observe an increase of efficiency with increase of both overlay size and storage buffer size. For all the cases, the total segment availability of the system rises which translates to a better upload bandwidth efficiency. More optimizations for bandwidth efficiency can be achieved by using lower level granularity for data transmission since we use segment level transmission in our simulator. We also plot the cumulative probability distribution as a function of percentage of upload bandwidth usage in Figure 6 for $b = 4$ and an overlay size of 6,000. The same pattern follows with various overlay sizes. We can notice from Figure 6, that the upload bandwidth usage is efficiently utilized and distributed among the participants with a maximum usage of 60% for $b = 4$ for an overlay size of 6,000. We feel that this is an area for further improvement by carefully analyzing

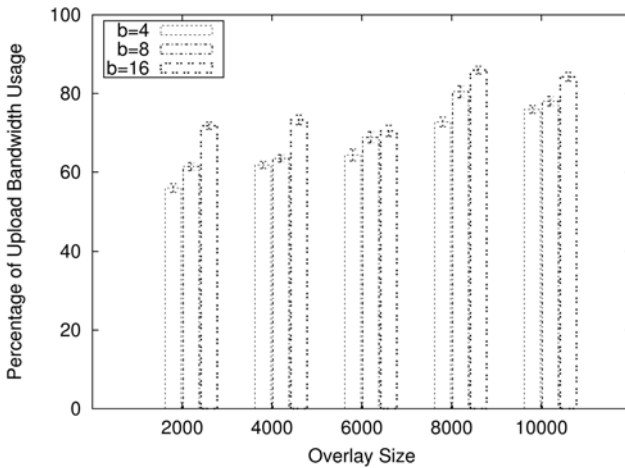


Fig. 5. Upload bandwidth usage efficiency

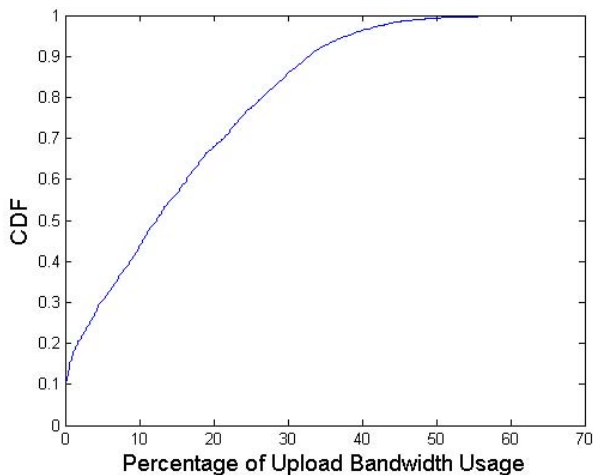


Fig. 6. Cumulative Distribution Plot for a function of % Upload Bandwidth Usage for $b=4$, $N=6,000$ peers

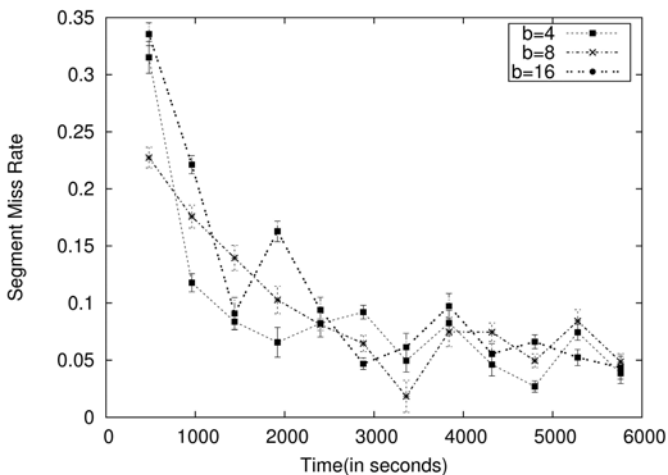


Fig. 7. System specific streaming performance at continuous time interval

the situation and optimizing the bandwidth usage to a greater degree which will help to improve the overall system performance.

5.4 Peer Churn/Departure

In this section we experiment the performance of *COCONET* in churn/failure conditions. We evaluate streaming performance during peer failure/departure. We adopt a metric known as Segment Miss Ratio(SMR) which is basically the

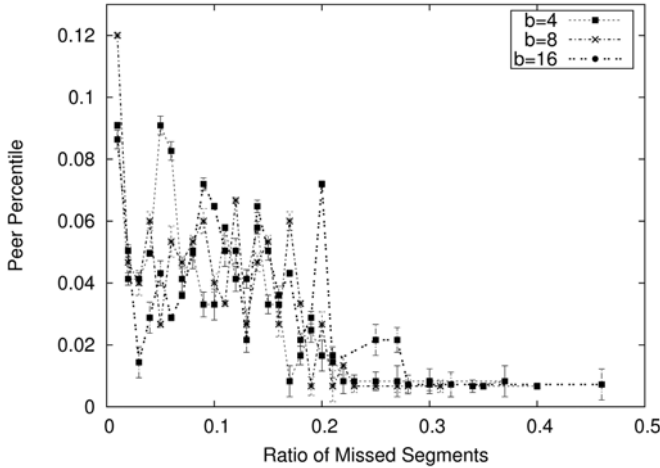


Fig. 8. Node specific streaming performance for each node in the overlay

number of segments that have not reached the playing buffer within playback deadline divided by the total number of segments that should have been played till that time. Initially we start with 10,000 peers and after a while when all the peers have started playing, we randomly kill a peer every 10 seconds till we have failed 30% of the initial population. Figure 7 plots the SMR value as an average of the whole system at specific time intervals for different values of b . We can observe that initially the failure impact is more but as time progresses, more peers join and more storage buffers come into the system which increase segment availability. We analyze peer specific performance in Figure 8 where we plot the percentage of missing segments for each node. It can be seen that most of the peers have a SMR less than 10% and very few peers with high SMR. Each *COCONET* peer employs a failure recovery scheme by removing dead entries from S_M during the download request process if the neighbor fails to reply within a certain period of time. This will help to avoid wastage of messages to dead peers. We do not employ separate heartbeat process for failure recovery since this process works well in our scheme with the added advantage of lesser overhead of control messages.

5.5 VCR Operations

In this section we study the effect of VCR operations such as random seek/FF/FB in *COCONET*. To simulate random-seek, we employ the same model from Section 5.4 where the peer failure events are replaced by random seek. Figure 9 plots the result which indicates that the SMR is initially very high but after a certain period of time is almost averaging around 5%. We also simulated fast forward VCR operation and plotted in Figure: 10 to study its effects. Again we used the same model from Section 5.4 with peer failure events replaced by

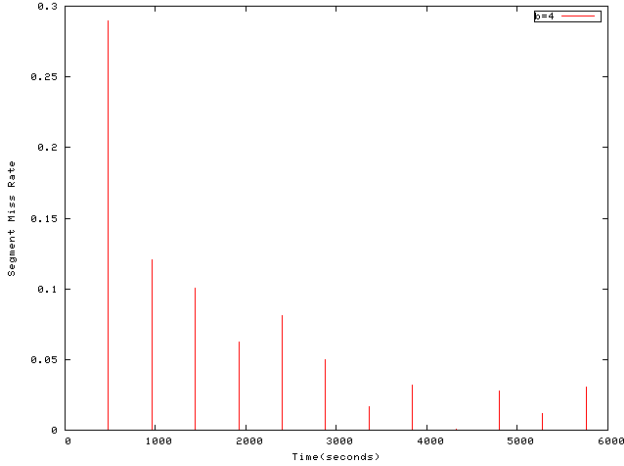


Fig. 9. Streaming performance for random seek operation

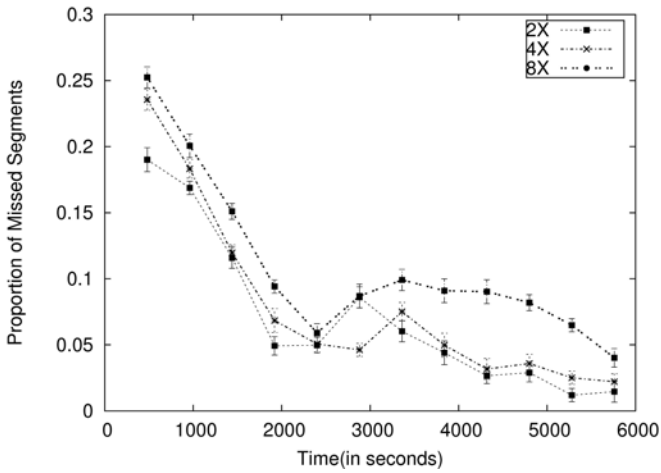


Fig. 10. Quality of streaming for fast-forward operation

FF. We experimented FF operation for different speeds such as 2X, 4X and 8X where essentially for 2X we play the same content quantity with double the speed and likewise. We can observe from the graph that 8X has the highest missing segments whereas 2X and 4X are within comparable ranges.

6 Conclusion

This paper proposes a novel way of organizing peers based on storage cache content where each peer contributes a part of storage to form a large distributed

cache which helps in alleviating server load to a greater extent by co-operative caching. Some of the most notable features of *COCONET* are high segment availability for any viewing patterns, uniform query load distribution, randomized content distribution pattern with uniform storage cache access pattern. As future work, we would like to: (a) deploy *COCONET* in PlanetLab for exercising its performance under real network dynamics, (b) employ certain predictive pre-fetching schemes to intelligently recover segments based on user viewing pattern, and (c) extend for multi-video scenario where the storage buffer can also be utilized to serve other peers watching different movies.

References

1. Huang, C., Li, J., Ross, K.W.: Can Internet Video-on-Demand be Profitable? In: Proceedings of ACM SIGCOMM, pp. 133–144 (2007)
2. Wang, D., Liu, J.: A Dynamic Skip List Based Overlay for On-Demand Media Streaming with VCR Interactions. *IEEE Trans. Parallel and Distributed Systems*, 503–514 (2007)
3. Do, T., Hua, K.A., Tantaoui, M.: P2VoD: Providing fault tolerant video on-demand streaming in peer-to-peer environment. In: Proc. of ICC, pp. 1467–1472 (2004)
4. Hefeeda, M., Habib, A., Botev, B., Xu, D., Bhargava, B.: Promise: Peer-to-Peer Media Streaming using CollectCast. In: Proceedings of ACM Multimedia, pp. 45–54 (2003)
5. Cheng, B., Jin, H., Liao, X.: Supporting VCR Functions in P2P VoD Services using Ring-Assisted Overlays. In: Proc. of the IEEE Intl. Conf. on Communications, pp. 1698–1703 (2007)
6. Vratonjic, N., Gupta, P., Knezevic, N., Kostic, D., Rowstron, A.: Enabling DVD-like Features in P2P Video-on-Demand Systems. In: Proc. of SIGCOMM Peer-to-Peer Streaming and IPTV Workshop (2007)
7. Kermarrec, A.M., Massoulie, L., Ganesh, A.J.: Probabilistic Reliable Dissemination in Large-Scale Systems. *IEEE Transactions on Parallel and Distributed Systems*, 248–258 (2003)
8. Zegura, E., Calvert, K., Bhattacharjee, S.: How to model an internet network. In: Proceedings of IEEE INFOCOM, pp. 594–602 (1996)
9. Huang, Y., Fu, T.Z.J., Chiu, D.M., Lui, J.C.S., Huang, C.: Challenges, Design and Analysis of a Large-scale P2P-VoD System. In: Proceedings of SIGCOMM, pp. 375–388 (2008)
10. Qiu, X., Wu, C., Lin, X., Lau, F.C.M.: InstantLeap: Fast Neighbor Discovery in P2P VoD Streaming. In: Proceedings of ACM NOSSDAV, pp. 19–24 (2009)
11. Cui, Y., Li, B.C., Nahrstedt, K.: oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks. *IEEE Journal on Selected Areas in Communication*, 91–106 (2004)
12. Castro, M., Druschel, P., Kermarrec, A.M.: SplitStream: High-bandwidth content distribution in a cooperative environment. In: Proceedings of SOSP, pp. 298–313 (2003)
13. Gou, Y., Suh, K., Kurose, J., Towsley, D.: P2Cast: Peer-to-Peer patching scheme for VoD service. In: Proceedings of WWW, pp. 301–309 (2003)

QShine 2009

**Session II – Multi-hop Wireless
Networks**

Opportunistic Multipath Routing in Wireless Mesh Networks

Jack W. Tsai^{1,2} and Tim Moors¹

¹ University of New South Wales, Sydney, Australia

² NICTA, Australia*

jackwtsai@gmail.com, moors@ieee.org

Abstract. This paper investigates combining opportunistic routing techniques with multipath routing for achieving reliability and timeliness in fast-changing network conditions. We present two approaches, WIMOP and DOMR, based on source routing and distributed routing, respectively. Instead of using broadcast packets as in most opportunistic routing work, we use unicast with promiscuous listening so that the reliability at each hop can be increased through retransmissions, while maintaining the broadcasting property required by opportunistic routing. We evaluate our work in NS2 against single path routing and MORE. Our results show that using the same amount of redundant data, our approaches were able to achieve better reliability than MORE. In addition, DOMR also has the advantage over WIMOP that it requires significantly less computational time.

Keywords: Wireless, mesh, multipath, routing, opportunistic.

1 Introduction

Wireless mesh networks have been used to provide cheap network connectivity in a range of scenarios. The main challenge for mesh networks, especially in outdoor applications, is to overcome unstable and unreliable links caused by the environment such as fading, moving vehicles and external interference. Compared to WLANs, these problems associated with the wireless medium are magnified due the multi-hop nature of wireless mesh networks. It has been observed in some testbeds that many links have loss ratios as high as 50% [1]. Recently, a new breed of routing protocols, based on opportunistic routing, has emerged to address such issues.

In traditional routing protocols, a next-hop is selected before the packet is forwarded towards the destination. On the other hand, opportunistic routing exploits the broadcast nature of the wireless medium such that nodes which overheard a packet transmission can also participate in the packet forwarding process, therefore

* NICTA is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

increasing the probability of a packet transmission being successful at propagating the packet towards the destination.

To date, the main focus and application of opportunistic routing has been on improving traffic throughput by reducing the medium access time. In this paper we investigate using opportunistic routing to improve performance for those applications that require reliability and timeliness instead. In opportunistic routing, depending on the receptions at each hop, packets can, and usually do take different paths to the destination. This is in some ways similar to multipath routing, though the choice of multiple paths isn't predetermined by the source. Our work is to build on existing opportunistic routing but also explicitly require the delivery to use multiple paths in the process. We have previously proposed using multipath routing as a means to achieve these goals [2]. In this work, multiple paths are calculated from the source node such that the interference between different paths (inter-path) and also between nodes within each path (intra-path) is minimised. This method requires an exhaustive search of the path space and thus is not easily scalable, and also relies on the accuracy and freshness of the link metric at the source node. Any short term link quality changes at the intermediate nodes are not taken into account.

Since our proposed work needs to adapt to fast changing network conditions, we first investigate a new link quality estimation technique that rapidly reflects the link quality by complementing ETX [3] with the number of link layer retransmissions measured from the successfully transmitted packets.

In this paper we investigate two approaches of applying opportunistic routing to multipath routing. The first builds on our previous work in source-based multipath routing. At each node, the link quality of the next-hop link specified in the source-routed packet is constantly monitored, so that when it deteriorates below a certain threshold, the node can choose to bypass the link by electing a neighbour node to capture the transmission and forward the packet using the neighbour's best available path. The main benefit of this addition is that we can overcome the potential inaccuracy in metric calculation at the source due to the lag in time in the exchange of link quality information.

Our second approach is a distributed opportunistic multipath routing protocol. At each hop, a list of candidate forwarders is calculated using link quality and interference information. Each candidate forwarder has a probability of forwarding that depends on the quality of the paths it provides to the destination. Therefore the amount of redundant data in the network is controlled by adjusting the forwarding probabilities. The proposed routing protocol has the following properties:

- Distributed
- Opportunistic and multipath
- Interference aware

The rest of the paper is organized as follows. In §2 we present the related work. In §3 the new link quality metric and the two approaches to opportunistic routing are presented. Simulation results from NS2 are presented in §4, and we conclude the paper in §5.

2 Related Work

Opportunistic routing exploits the broadcast nature of the wireless medium by allowing more than one neighbour to participate in the packet forwarding process. The concept of opportunistic routing was first proposed in ExOR [4]. The two main issues in opportunistic routing are forwarder selection and coordination rules. Since it is clearly detrimental to involve all neighbouring nodes in packet forwarding (for example, using nodes farther from the destination), forwarder selection is used to choose a set of neighbours that will provide the best forwarding performance. Coordination rules help decide which of the forwarders that have successfully received a transmission should forward the packet. This is to prevent unnecessary transmissions caused by forwarding packets that are likely to have been forwarded by other nodes already, which waste bandwidth and create interference.

In ExOR the sender computes a forwarder list by ranking the neighbours in terms of their ETX to the destination and picking those with smaller values than itself. It enforces coordination between the forwarders by using a strict packet scheduler in the MAC layer; each forwarder is given a time slot according to its priority and can only transmit during that slot. During a transmission, other forwarders listen in and record the packets that are being sent. This information is passed on where possible so that packets already transmitted by one forwarder are dropped by the rest.

The main problem with ExOR is that it requires a customised MAC in order to schedule packets, which increases hardware cost and restricts deployment. In contrast, MORE [5] proposed using randomness provided by network coding to eliminate the need for a scheduler and thus can be used with 802.11. This works by requiring nodes to transmit coded packets, which are linear combinations of multiple packets. A forwarder receives a coded packet and decides if it contains new information that it has not already received, if so it will forward a new linear combination of the received coded packets. When the destination receives enough coded packets to reconstruct the original packets, it immediately acknowledges the source to initiate the transmission of a new batch of packets. Because the data transmitted by each forwarder is a linear combination of received packets with random coefficients, the probability of nodes transmitting the same information and wasting bandwidth is greatly reduced, therefore a scheduler is no longer needed.

The forwarder selection in both ExOR and MORE considers only the ETX cost of each forwarder to the destination. While this is simple, it might not be optimal in achieving reliability. [6] addresses the least-cost opportunistic routing problem by proposing an algorithm that assigns and prioritises the set of candidate relays (forwarders) so that the cost of forwarding a packet to the destination is minimised. In our work we propose a similar forwarder selection that considers spatial diversity and also explicitly allow multiple copies of a packet to be forwarded.

The opportunistic protocols mentioned so far all focus on using broadcasts to forward packets to the destination, with some requiring link layer scheduling. However, unlike unicast, broadcast does not use retransmissions and collision detection (RTS/CTS). As a result, broadcast based routing might not perform well in terms of reliability when links are very unreliable. In contrast, we propose a unicast-based opportunistic routing that at least ensures reliability to a certain degree.

3 Protocol Description

3.1 Link Quality Estimation

ETX and similarly derived broadcast link quality metrics have recently been shown to be poor indicators of the real link quality experienced by data traffic in some cases. One problem is that ETX measures the performance at the receiver of a link without considering the exponential back-off time incurred by channel contention at the sender. The lack of RTS/CTS for broadcast traffic also causes inaccurate estimations due to the hidden terminal problem.

As a first step to improve link quality estimation, we note that the 802.11 MAC layer provides useful information about the current link condition, such as link rate, SINR, retransmission counts, etc. We are particularly interested in the number of retransmissions actually performed to forward a packet to a neighbouring node, which is closely related to the ETX metric concept. The advantage of using the retransmission counts to gauge the link quality is that it is measured on unicast traffic, which eliminates the problem with broadcasted probes as described before. In addition, it does not add to traffic overhead like probe-based metrics do. The main problem with retransmission counts, however, is that it requires active traffic over the link in order to be of any significance. To solve these problems, we propose to combine a probe-based metric with local MAC layer retransmission counts to get the best of both worlds.

The main use of the ETX base metric is when there is little or no traffic on the networks, it has been shown that under these conditions ETX base metrics give very accurate estimates of link qualities [7]. Thus the new metric should have a component of ETX whose weight is inversely proportional to the data traffic on the link:

$$ETX-R = \left(ETX \times \left(1 - \frac{t_{i,j}}{b_{i,j}} \right) + r_{ij}(T) \times \frac{t_{i,j}}{b_{i,j}} \right) \quad (1)$$

where

$r_{ij}(T)$ is the average number of link layer retransmissions required for a successful unicast transmission in the last T seconds

$t_{i,j}$ is the transmission rate (load) on link i,j

$b_{i,j}$ is the link rate of link i,j

3.2 Load-Aware Path Metric

In our prior work [2] we proposed a multipath routing algorithm based on interference minimisation. The source node computes the paths that are used for forwarding traffic to a destination using a WIM score. The WIM score reflects the degree of interference between different paths in a multipath set as well as individual path quality. The main component in the WIM score is the interference cost for a link i,j operating on channel c in a network N ,

$$LI_{ij}(c, N) = ETT_{ij}(c) * |E_{ij}(c, N)| \quad (2)$$

where ETT [8] is the expected transmission time for a packet of size S , derived from ETX and the link bandwidth B ,

$$ETT_{ij}(c) = ETX_{ij}(c) * S/B, \quad (3)$$

and $|E_{ij}(c, N)|$ is the number of mutual interference sets (set of links in which only one link can successfully transmit at a time) affected by transmission on link i, j .

The original LI calculation assumes that all links are active, while in many real-life application traffic are often bursty and links can be idle for periods of time. Consequently the original LI calculates the worst-case interference. Therefore in this paper we modify link interference estimation to include the load of a link. The load-aware link interference is:

$$LLI_{ij}(c, N) = ETT_{ij}(c) * LF_{ij} * S_{ij}(c, N), \quad (4)$$

where $S_{ij}(c, N)$ is the number of nodes that link ij interferes with.

The WIM score for a path set P is a weighted sum of the aggregated link interference of P and that of other nodes in the network ($N-P$), using a weight factor β .

$$PIC = \sum_{ij \in p} LLI_{ij}(c, P) \quad (5)$$

$$NIC = \sum_{ij \in p} LLI_{ij}(c, N - P) \quad (6)$$

$$WIM = \beta * NIC + (1 - \beta) * PIC \quad (7)$$

3.3 Unicast-Based Opportunistic Routing

Unlike the conventional approach to opportunistic routing, which uses broadcast to forward traffic towards the destination, our protocol adopts a unicast-based approach similar to that in [9]. Instead of sending packets in broadcast mode, the sender uses unicast to send packets to a primary recipient, while other nodes in the vicinity listen promiscuously and opportunistically capture packets. The main motivation behind our approach is to ensure reliability. In the 802.11 MAC broadcast mode there is no acknowledgement so the sender does not retransmit. Consequently reliability suffers. Broadcast-based opportunistic routing has been shown to improve data throughput and network capacity [4] [10], but link layer retransmission may be a better way to ensure reliability than having to implement acknowledgements above the link layer. Therefore in our proposed framework, each node listens in on wireless transmissions regardless of whether they are broadcasts or unicasts.

3.4 Approach One: Source Multipath Routing (WIMOP)

The source-based multipath routing algorithm we proposed in [2] relies on the accuracy of the metric at the time of the route computation to perform well. However in some networks even if the topology is static, link quality may vary greatly due to the environment. In these cases the time required for the link state algorithm to

disseminate link quality information across the network may be longer than the coherence time of the channels, and therefore the link metrics do not reflect well the actual link qualities. To overcome this problem we propose using opportunistic routing at nodes along the paths when a link deteriorates.

Each node monitors the number of retransmissions used (r_{ij}) in the transmission of packets to each of its neighbour. A long term and a short term exponential moving average of the retransmission counts are computed for each link. These averages are used to help identify short term link deterioration. The source sends out data as before using source routing, at each intermediate node the quality of the next-hop link is examined. When the short term average r_{ij} drops below the long term average by more than a threshold value, indicating that the instantaneous link quality is below expectation, the node will find an alternative next-hop to collaborate in the forwarding of the packet. The node considers the total cost of delivering the packet via each of its neighbours without forming a loop. The neighbour with the best cost is flagged in the packet header as a *collaborator* to which data is forwarded using unicast transmissions until the short term r_{ij} of the next-hop link indicated in the source route moves back above the threshold. In addition, if the original next-hop node captures a packet sent to the collaborator, it will still need to forward the packet. In other words, the nodes in the source routes will opportunistically forward packets that are captured.

When the collaborator captures a packet, it calculates the best path to deliver the packet to the destination using the WIM calculation on the path candidate and existing multipath packets already indicated in the packet. This is done so the new path is selected to avoid interference with other paths delivering the same packet.

The detail is described as follows in Algorithm 1 and is illustrated in Fig. 1.

Algorithm 1

$C_{i,d}(j)$ is the path cost from node i to destination via node j
 $T_l > T_s, T_s, T_l$ are the intervals for retransmission counts collection

- 1 If $r_{ij}(T_s) > r_{ij}(T_l)$
- 2 do for all node n in neighbours
- 3 find $n/$ such that $C_{i,d}(n/) = \min(C_{i,d}(n))$
- 4 do flag packets to s to indicate $n/$ as collaborator
- 5 unicast packet to collaborator
- 6 continue until $r_{ij}(T_s) < r_{ij}(T_l)$

Limiting the Number of Duplicates in the Network

When a collaborator is used, if the original next-hop also captures the packet successfully, the number of duplicates of the packet in the network is increased by one. Therefore the maximum number of duplicates that may be generated along a path is the length of that path. In order to limit the number of duplicates in the network and therefore save bandwidth, a packet that has been already forwarded by a node is dropped. In addition, if the packet has not been forwarded by a node but has been seen transmitting by the next-hop node, i.e., the next-hop node has forwarded the packet already, the packet is also dropped.

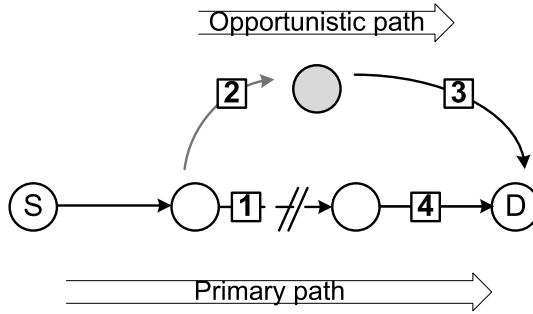


Fig. 1. Illustration of opportunistic routing when link deteriorates. 1. The local sending node detects short term deterioration of next-hop link and tags a collaborator in the packet header 2. The collaborator receives the packet via opportunistic listening 3. The packet is forwarded towards the destination by the collaborator 4. If the next-hop node on the primary path receives the unicast packet it will forward the packet as per usual.

3.5 Approach Two: Distributed Opportunistic Multipath Routing (DOMR)

Compared to WIMOP, DOMR provides a distributed approach to opportunistic multipath routing. The omission of source routing and exhaustive search of paths means that DOMR has a lower requirement for computational power and thus better scalability.

In DOMR each node keeps information about every flow it has participated in during the last T_{out} seconds. The timeout value is used to purge information of stale flows. Two parameters are set by the source: N_p , the redundancy factor, and N_f , the number of potential paths.

We classify forwarders into *primary* and *opportunistic*, each with a different forwarding behavior.

Forwarder – Primary

The *primary forwarders* are the forwarders along the primary path computed by the source. At each primary forwarder, the best next-hop may not be the same as the one identified by the source due to fluctuation in link qualities. Therefore a primary forwarder should dynamically switch to a better link, this is done using algorithm 1.

Forwarder – Opportunistic

Opportunistic forwarders are identified by the primary forwarder in the packet header. When an opportunistic forwarder captures a packet it will forward the packet using its best metric path with a probability given in the forwarder list. The forwarded packet will contain a new forwarder list and forwarding probabilities computed using local information. By changing N_p and N_f as a packet travels downstream, we can control the degree of redundancy and whether to further branch out packet transfer ($N_f > 1$).

Each node prepares routing by listing forwarders whose path costs to the destination are less than that of itself, similar to ExOR. The path with the best

aggregate ETT is identified as the primary path. A WIM score is then computed between each forwarder's best ETT path and the primary path. The forwarders list is then ranked according to the WIM scores and truncated to leave the top N_f forwarders. Each forwarder i is given a forwarding probability according to the proportion of its WIM score (WIM_i) with regard to the total WIM score:

$$p_i = \frac{1/WIM_i}{\sum 1/WIM_i} \times N_p, \quad (8)$$

In other words, a forwarder is more likely to forward a packet if it has a lower interference path to the destination when compared to the primary path. Assuming every forwarder successfully captures the packet, the expected number of forwarders that will forward the packet is N_p , which satisfies the redundancy requirement. In order to reduce computational time, the probabilities are recomputed only if the metric of a link has changed by more than a threshold. To ensure reliability on the primary path, if p_i of the primary forwarder is less than 1, it is adjusted to 1 and then p_i of the opportunistic forwarders re-calculated using $N_p - 1$ instead of N_p . Also if $N_p = N_f$ then some forwards may have a p_i of greater than 1. In this case we could either allow forwarders to perform more than 1 transmission of the same packet, or setting the forwarding probabilities of all forwarders to 1.

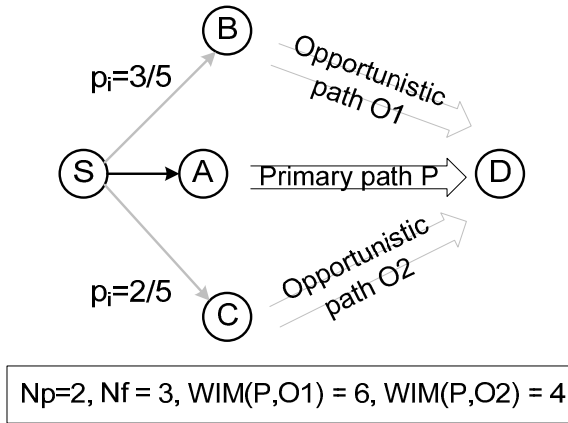


Fig. 2. DOMR forwarding probability calculation. Node A is the next-hop node on the primary path from S to D. S calculates the WIM scores for path sets (P,O1) and (P,O2) and then determines the forwarding probability for node B and C. A packet is sent from S to A and identifies B and C as opportunistic forwarders, each has a forwarding probability of 3/5 and 2/5, respectively.

By using different N_p and N_f , the source can control the number of paths a packet will take and also how frequently each path is used. For example, setting N_p and N_f both to 3 will allow 3 paths each with forwarding probability of 1 (if p_i cannot be

greater than 1). While using a smaller N_p of 2 results in the same number of paths, but a lower volume of redundant traffic.

The forwarder list is included in the packet header along with other information such as N_p and N_f , and the estimated path metric to the destination from the next hop. The estimated path metric is calculated during the computation of the routing table and therefore does not incur additional computation time. The packet is then sent using unicast to the next hop.

Fig. 2 gives an example on the forwarding probability calculation in DOMR.

4 Evaluation

We evaluate our opportunistic multipath routing protocol using the NS-2 simulator. In this section we present the simulation results.

4.1 Simulation Setup

We used a modified version of NS 2.33 simulator. We have modified the link layer component to support multiple radios and multiple orthogonal channels. We use the Optimized Link State Routing (OLSR) protocol [11] to provide the basic link-state exchange framework, the Topology Control (TC) messages in OLSR now carry link information of all the interfaces in each node, rather than that of the main interface only. This allows each node to have full information of the connectivity in the network.

The network topology is generated by placing 100 nodes randomly over a 2km x 2km area. The distance between any two nodes is at least half the transmission range, and every node has at least one neighbour with which it can communicate. The topology is fully connected. The minimum distance requirement between nodes ensures that the topology is evenly spaced and there is an upper bound on the node density. Each node is equipped with two 802.11b radios tuned to orthogonal channels such that we can assume that there is no interference between the channels. The channel assignment is static and redundant, i.e. every node operates on the same pair of channels. Instead of the frame capture model available in older versions of NS, we use the new physical interface extension that supports additive SINR. The transmission rate on each link is set using the distance/rate relationship defined in Table 1. To model loss and fading in NS2 we apply the Gilbert-Elliot loss model with an average loss ratio of 0.5.

Table 1. Distance/Rate Relationship of Radios

Distance (m)	60	120	180	250
Rate (Mbps)	11	5.5	2	1

4.2 Link Quality Estimation

First we evaluate ETX-R against ETX. We performed simulations to test the performance of the metrics under interference from neighbouring nodes. 5 topologies

of 50 nodes each are randomly generated. For each topology we performed 5 simulations, each consists of one main data flow and three interference flows. The traffic rate of the main flow is increased until the maximum throughput can be established. The sending rate of the interference flows were increased in 50 pkt/s increments from 0 pkt/s to 100 pkt/s. Two sets of simulations were performed, one using ETX and the other using ETX-R as the link metric. The results in Fig. 3 show that ETX-R was able to select higher-reliability paths. As discussed before, the performance of ETX depends on the broadcast interval of probe packets as well as the rate of link state information exchange. ETX-R is able to reflect quicker the change in link quality if there are enough transmissions taking place. In order for ETX to reach the same responsiveness in our scenario, the intervals would need to be much smaller than 1 second, which would result in an unacceptable increase in overhead traffic.

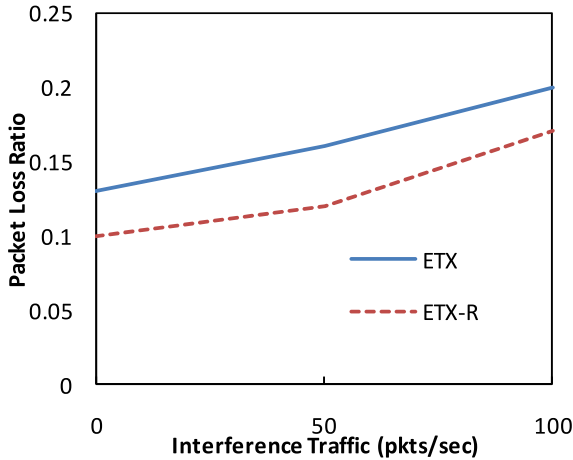


Fig. 3. Performance comparison between ETX and ETX-R

4.3 WIMOP

In this section we examine the performance improvement achieved as we add dynamic routing and opportunistic receiving to source-based multipath routing.

Using the opportunistic routing techniques described in §3.3, we modified our prior work in WIM [2] and compared it with the original version. The intervals, T_s and T_l , for calculating the transmission count moving averages in WIMOP-S, are set to 2 seconds and 0.5 seconds, respectively. We performed simulations using 2 and 3 paths.

Fig. 4 shows the resulting packet loss rate and end-to-end delay with a range of link layer retransmission limits. The addition of opportunistic routing to source multipath routing improved the packet loss ratio and lowered end-to-end delay. The lowered

delay, in particular, is due to the fact that redirecting packets avoids incurring a large number of retransmissions at the problem link. Table 2 shows the proportion of packets that were transmitted using alternate hops due to temporary link deterioration on the primary hop.

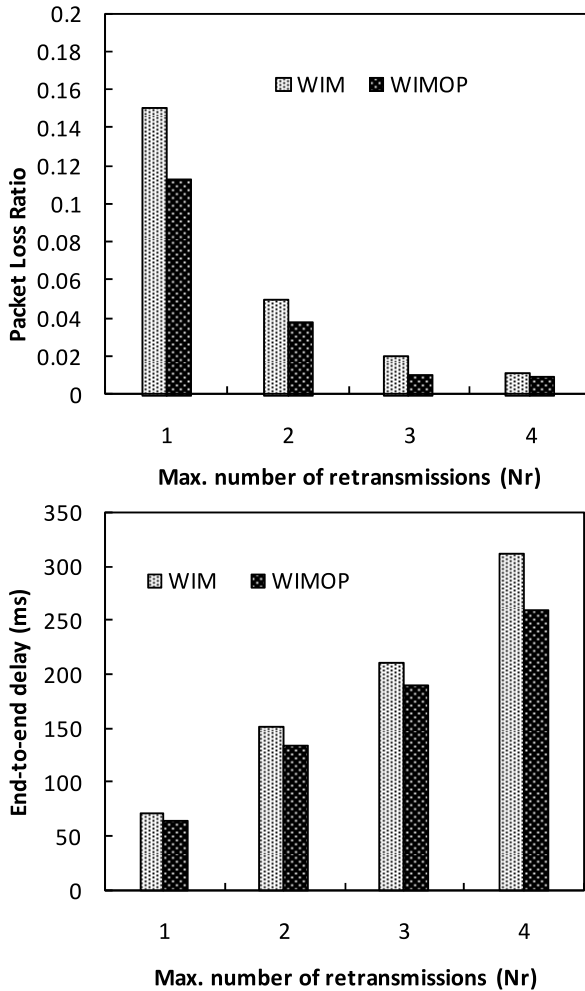


Fig. 4. Performance improvement of opportunistic source routing using WIM

Table 2. Proportion of packets rerouted

Nr	1	2	3	4
Rerouted %	7.1	9.6	14.4	21.3

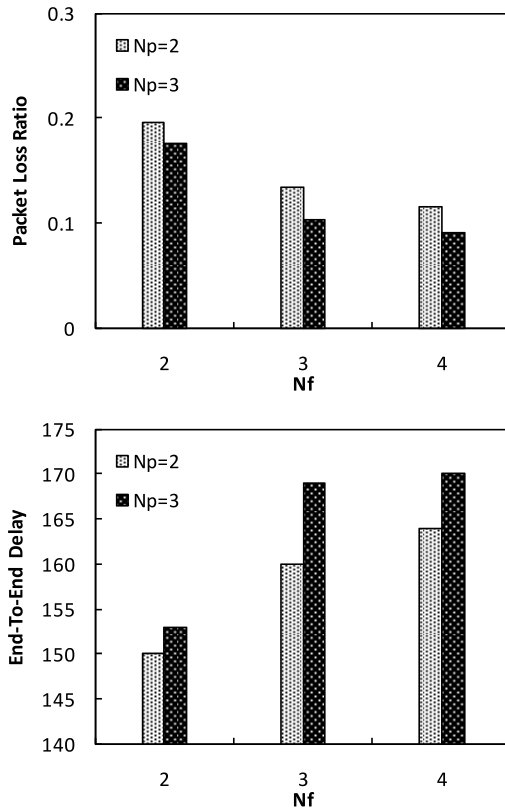


Fig. 5. Performance using different redundancy factor, N_p , and number of paths, N_f

4.4 DOMR

In this section we evaluate the performance of distributed opportunistic multipath routing based on our algorithm described in §3.4. The performance of DOMR is evaluated against single path routing using WCETT [8], MORE, and sourced-based opportunistic multipath routing (WIMOP).

Since MORE is a reliable file transfer protocol, it continues to resend packets to the destination until all packets in a batch are received and acknowledged. In order to evaluate its general performance as a routing protocol, we set a limit to the number of packets the source can send for a batch of packets to N_p times the number of packets, the redundancy factor used in DOMR. After the limit is reached, the source will move on to the next batch.

First we investigate the effect of varying the degree of redundancy, N_p , and the path diversity, N_f . Fig. 5 shows that an increased redundancy improves the packet loss

ratio of the data transfer at the cost of a slight increase in delay. Fig. 5 also shows that, provided with the same degree of redundancy, increasing path diversity allows more opportunistic listening and forwarding to take place and results in better reliability.

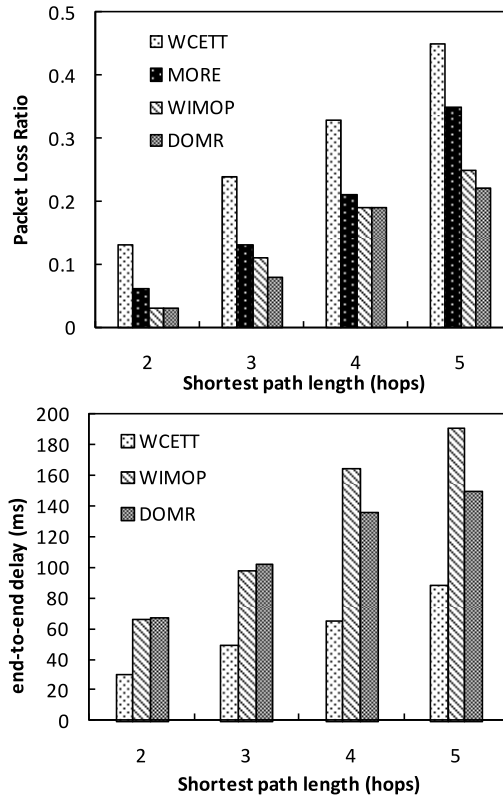


Fig. 6. Opportunistic routing performance comparison

Fig. 6 and Fig. 7 show the result of the simulations. Both WIMOP and DOMR were able to achieve better reliability than MORE. DOMR also has a slight advantage over WIMOP in both reliability and end-to-end delay. The low delay achieved by using WCETT compared to others is due to the fact that traffic is forwarded along the best metric path, however the lack of redundancy makes the packet loss ratio significantly higher. The delay for MORE is not included in the comparison since packets are coded and transmitted in batches, resulting in large overall delay and making calculation and comparison of per-packet delay difficult.

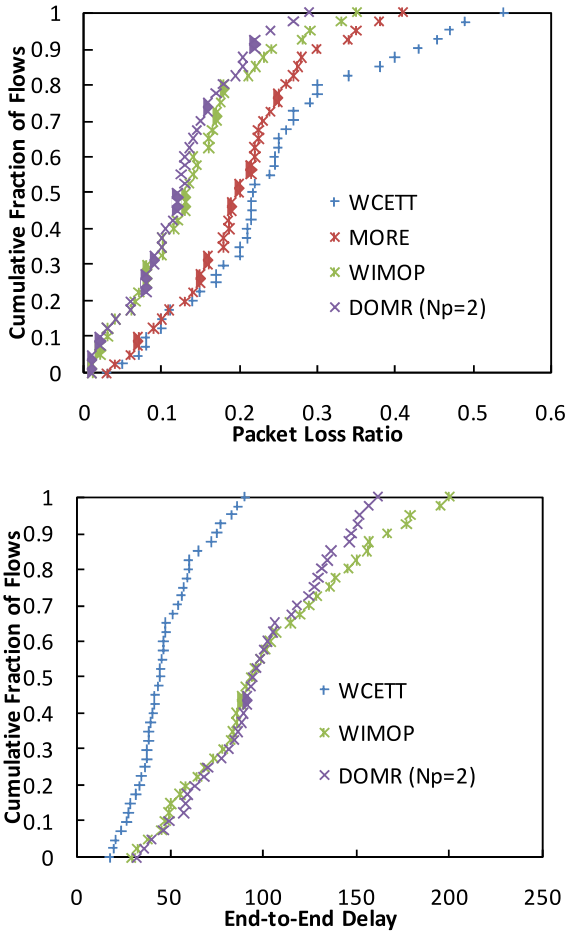


Fig. 7. Cumulative Distribution Function of Traffic Flows Performance

5 Conclusion

This paper presented two opportunistic multipath routing algorithms. The first is a modification to our prior work using source-based multipath routing. The addition of opportunistic routing techniques enables the routing to dynamically bypass links which have deteriorated since the calculation of routes at the source. The second of our proposed algorithms is completely distributed, and relies on varying each potential forwarder's forwarding probability to limit the number of duplicates in the network. We also evaluated using local link layer retransmission count to improve the accuracy of link quality estimation. Simulation results in NS2 showed that the addition of opportunistic routing further improved the reliability of source-based

multipath routing. Furthermore, both approaches were able to achieve better reliability than MORE.

References

- [1] Aguayo, D., Bicket, J., Biswas, S., Judd, G., Morris, R.: Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev.* 34, 121–132 (2004)
- [2] Tsai, J.W., Moors, T.: Interference-aware Multipath Selection for Reliable Routing in Wireless Mesh Networks. In: *MeshTech 2007*, pp. 1–6 (2007)
- [3] De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: *9th Ann. Int'l Conf. on Mobile Computing and Networking*, San Diego, CA, USA (2003)
- [4] Biswas, S., Morris, R.: ExOR: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.* 35, 133–144 (2005)
- [5] Chachulski, S., Jennings, M., Katti, S., Katabi, D.: Trading structure for randomness in wireless opportunistic routing. In: *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, Kyoto, Japan. ACM, New York (2007)
- [6] Dubois-Ferriere, H., Grossglauser, M., Vetterli, M.: Least-Cost Opportunistic Routing. In: *Forty-Fifth Annual Allerton Conferences*. Allerton House, UIUC, Illinois, USA (2007)
- [7] Subramanian, A.P., Buddhikot, M.M., Miller, S.: Interference aware routing in multi-radio wireless mesh networks. In: *2nd IEEE Workshop on Wireless Mesh Networks, WiMesh 2006*, pp. 55–63 (2006)
- [8] Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: *Int'l Conf. on Mobile Computing and Networking*, Philadelphia, PA, USA, pp. 114–128 (2004)
- [9] Katti, S., Katabi, D., Hu, W., Hariharan, R., Médard, M.: The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments. In: *Forty-Third Annual Allerton Conference on Communication, Control, and Computing* (2005)
- [10] Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., Crowcroft, J.: XORs in the air: practical wireless network coding. *SIGCOMM Comput. Commun. Rev.* 36, 243–254 (2006)
- [11] Optimized Link State Routing Protocol (OLSR),
<http://hipercom.inria.fr/olsr/>

Gateways Congestion-Aware Design of Multi-radio Wireless Networks

Djohara Benyamina¹, Abdelhakim Hafid¹, and Michel Gendreau²

¹ LRC, University of Montreal, Montreal, Canada

² CIRRELT, University of Montreal, Montreal, Canada

{benyamid, ahafid, michel.gendreau}@iro.umontreal.ca

Abstract. In Wireless Mesh Networks (WMNs), traffic is mainly routed by WMN Backbone (WMNB) between the mesh clients and the Internet and goes through mesh gateways. Since almost all traffic has to pass through one of the MGs, the network may be unexpectedly congested at one or more of them, even if every mesh router provides enough throughput capacity. In this paper, we address the problem of congestion of gateways while designing WMNs. We propose a simultaneous optimization of three competing objectives, namely network deployment cost, interference between network channels and congestion of gateways while guaranteeing full coverage for mesh clients. We tailor a nature inspired meta-heuristic algorithm to solve the model whereby, several trade-off solutions are provided to the network planner to choose from. A comparative experimental study with different key parameter settings is conducted to evaluate the performance of the model.

Keywords: Wireless mesh network design problem, Multi-objective model, simultaneous optimization, Congestion of gateways, Meta-heuristic method.

1 Introduction

The success of the Wireless Mesh Network (WMN) technology has caused a paradigm shift in providing high bandwidth network coverage to users. The Wireless Mesh Network Backbone (WMNB) consists of mesh routers (MRs) interconnected with each other through point-to-point wireless links to provide connectivity to mesh clients (MCs). MRs responsible for providing internet access to clients are called access points (APs) while other more expensive MRs, that are equipped with a gateway capability through which they interface with Internet, are called mesh gateways (MGs).

WMNs are highly reliable, scalable, adaptable and cost-effective. They are already pervasive in many diverse environments, such as home networking, enterprises, and universities. Nevertheless, users experience a number of problems, such as intermittent connectivity, poor performance and lack of coverage [1]. In Multi-Radio Multi-Channel (MR-MC) networks, MRs are equipped with multiple network interfaces, thus allowing simultaneous communications over orthogonal channels. However, since the number of available orthogonal channels is limited, interferences happen causing network performance degradation. A proper WMN design is a

fundamental task; if addressed carefully it can considerably improve the network efficiency in terms of coverage, throughput, delay and capacity.

Basically, the design of WMNs involves deciding where to install the network nodes (given a set of candidate locations), which type of nodes to select (AP, MG or simple MR), how many of these nodes to install, and which channel to assign for each node interface, while guaranteeing users coverage, wireless connectivity and traffic flows at minimum cost.

Exploiting the trade-offs among network deployment cost, network throughput, and congestion level of gateways, we propose in this study, a new approach to address the problem of WMNs design. Indeed, minimizing the cost requires stingy resources utilization (deploying fewer routers and/or gateways) which impacts the network performance. With few routers deployed, the traffic is routed on longer paths to get to its destination, thus increasing communications delays. With few gateways deployed, congestion may happen (since all traffic traverses gateways to and from the internet) affecting network throughput. Conversely, deploying more resources (higher deployment cost) helps providing shorter paths and less congested gateways; however, this may cause high interference levels and thus degrade network performance. In fact, optimizing one of these criteria will affect/undermine other(s) criteria(s). Therefore, it is difficult, if not impractical, to have a solution that is optimal in all criteria.

In this paper, we define a multi-objective optimization model that minimizes the network deployment cost, maximizes the network throughput (by minimizing the network interference level), minimizes congestion level of gateways, and guarantees a full coverage to mesh clients.

WMN design problem belongs to the set of Multi-commodity capacitated network design problems (MCNDPs). They are known to be hard combinatorial optimization problems for which several solution strategies have been developed. A number of these strategies involve the relaxation of some problem constraints and the strengthening of the model through the addition of valid inequalities [2]. In this study, we propose a multi-objective approach to search for the near-optimal set of non-dominated planning solutions. This set of trade-off solutions is very much desired by engineers who prefer to have several solutions in hand before taking decisions. An evolutionary population-based multi-objective algorithm based on particle swarm optimization (PSO) is developed to solve the proposed model.

Related research has mainly focused on the problem of performance improvement in order to effectively use WMNs; however, most of existing solutions assume a priori fixed topologies [3], [4], [5], [6]. Indeed, the design of a WMN is still in its infancy and many challenges remain open. Some studies [7],[8] consider topologies where gateways are fixed *a priori*, while others [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] attempt to optimize the number of gateways given a fixed layout of mesh routers. Very recent contributions in, [19], [20] however, propose WMN planning schemes where the locations of routers and gateways are not fixed. Nevertheless, all these studies consider in a way or another minimization of a single objective based on the deployment cost. We also stress the fact that users' coverage is not considered in [19] while QoS requirements, such as delay and throughput are not considered in [20]. None of these approaches tackle the issue of gateways congestion level in their problem formulation.

The key contributions of the paper can be summarized as follows: (1) a *novel* multi-objective optimization model; and (2) A meta-heuristic algorithm to resolve the model for real-size networks. To the best of our knowledge, there has been so far no real attempt to model WMN design problems using a pure multi-objective approach. The only work worth mentioning is presented in [11]; it concerns only gateways placement problem where locations of other mesh nodes are known *a priori*. Bing et al. [11] use a multi-objective approach but then aggregate the many objectives into a single one representing a weighted sum of objectives value. This is a classical approach to handle Multi-Objective Problems. However, the biggest problem with this aggregate approach is the inability to find solutions in non-convex fronts [21]. Moreover, the setting of the relative weights for the different objectives is subjective and often leads to favoring some and penalizing others.

The remainder of the paper is organized as follows. Section 2 defines the WMN planning problem and presents the mathematical formulation of the problem solution. The solution approach and the population-based algorithm are described in Section 3. Section 4 evaluates the proposed WMNs planning approach and Section 5 concludes the paper.

2 Network Model and Problem Formulation

Let I be the set of positions of traffic concentrations in the service area (Traffic Spots: TSs) and L the set of positions where mesh nodes can be installed (Candidate Locations, CLs)

The planning problem aims at:

- Selecting a subset $S \subseteq L$ of CLs where a mesh node should be installed so that the signal level is high enough to cover all TSs. This will constitute the set of APs.
- Defining the gateways set by selecting a subset $G \subseteq L$ of CLs where the wireless connectivity is assured so that all traffic generated by TSs can find its way to reach the nodes in G .
- Maintaining the cardinalities of G and S small enough to satisfy the financial and performance requirements of the network planner.

2.1 Network Model

In order to describe the problem formally we introduce the following notation:

Given n TSs and m CLs, let $I=\{1,\dots,n\}$ and $L=\{1,\dots,m\}$. In the following, unless otherwise stated, i and j belong to I and L respectively. The cost associated to installing a mesh node j is denoted by c_j , while p_j is the additional cost required to install a gateway at location j . d_i is the traffic generated by TS i . u_{jl} is the traffic capacity of the wireless link between CLs j and l . v_j is the capacity of the radio access interface of CL j . The coverage and connectivity parameters are given respectively by the binary variables a_{ij} and b_{jl} . a_{ij} takes the value 1 whenever TS i is covered by a mesh node in CL j . The parameter b_{jl} indicates whether CLs j and l can be connected via a wireless link. We define other 0-1 decision variables x_{ij} , g_j , t_j in our formulation

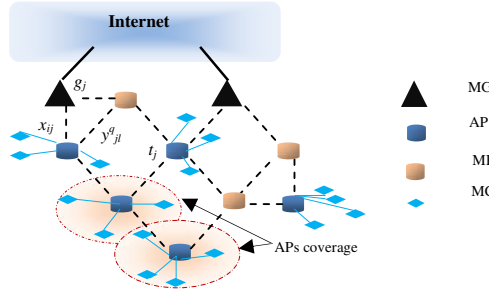


Fig. 1. WMN Planning Problem. A MC covered by many APs is assigned to only one AP.

(see Fig. 1). The variable x_{ij} takes the value 1 if TS i is assigned to CL j while t_j (g_j) is set to 1 if a router (a gateway) is installed in CL j .

We consider a multi-radio multi-channel WMN and we suppose initially that the mesh nodes operate using the same number of radios R , each with k channels, ($k > R$) and $k \in C$, where $C = \{1, \dots, c\}$ and c can be at most 12 orthogonal channels if IEEE802.11a is used. Extra installation variables are added: $z^q_j = 1$ if a mesh node is installed in CL j and is assigned channel q , $q \leq k$, $y^q_{jl} = 1$ if a wireless link from CL j to CL l using channel q ($q \leq k$) exists. Finally, we define the flow variables f^q_{jl} and F_j . the first variable denotes the traffic flow routed from CL j to CL l using channel q , the second is the traffic flow on the link between a gateway j and the Internet.

We represent our WMN as an undirected graph $G(V, E)$, called a connectivity graph. Each node v represents a mesh node which can be AP, MR or MG. The neighborhood of v , denoted by $N(v)$, is the set of nodes residing in its transmission range. A bidirectional wireless link exists between v and every neighbor u in $N(v)$ and is represented by an edge (u, v) . The maximum degree of G denoted by Δ is bounded by the number of radio interfaces of each node v . Table 1 summarizes the notation used in the problem formulation.

Table 1. List of Main Parameters/Variables Used in Model Formulation

<i>Par./V</i>	<i>Description</i>
n	Number of Traffic Spots (TSs)
m	Number of Candidate Locations (CLs)
d_i	Traffic generated by TS _{i}
u_{jl}	Traffic capacity of wireless link (CL _{j} , CL _{l})
v_j	Capacity limit for AP radio access interface
c_j	A device cost installation
p_j	A gateway additional cost installation
R	Number of radio interfaces
k	Number of channels
a_{ij}	Coverage of TS _{i} by CL _{j}
b_{jl}	Wireless connectivity between CL _{j} and CL _{l}
t_j	Installation of a device at CL _{j}
g_j	Installation of a gateway at CL _{j}

Table 1. (continued)

x_{ij}	Assignment of TS _i to CL _j
z_j^q	Installation of a device at CL _j , assignment of channel q, q<k
y_{jl}^q	Establishing a wireless communication on q Between (CL _j ,CL _l)
f_{jl}^q	Flow on channel q between (CL _j ,CL _l)
F_j	Flow on the wired link from CL _j to Internet
N_{jl}	Set of links interfering with the link y_{jl}^q

2.2 Problem Formulation

The planning approaches in [19], [20] focus on one or two optimization criteria at the expense of other network characteristics. Kodialam et al. [22] report that there exist multiple design criteria for WMNs; their proposal allows optimizing a single objective function at a time but no generic method for dealing with the multiple metrics is provided. The work in [12], propose a model (within a tool) to measure the performance of a designed WMN prior to its deployment. The main idea is to define: (1) a set of metrics that work as evaluation criteria for WMNs; and (2) a weighted combination of the metrics for a simultaneous use of multiple evaluation criteria in WMNs optimization. In the following, we describe the main criteria considered in our problem formulation.

Deployment Cost. Minimum installation cost is a fundamental issue in deploying WMNs. Increasing the number of MGs may increase the network throughput and may lead to a smaller number of gateway bottlenecks. Thus, we need to determine the right places of APs and MGs that result in: (1) a minimum number of APs that provides full coverage; and (2) a minimum number of MGs that provides enough throughput while satisfying QoS constraints. The first objective function to optimize computes the total cost of the network including installation cost c_j and additional MGs installation cost p_j .

$$\text{Min} \sum (c_j t_j + p_j g_j). \quad (1)$$

Network Throughput. Because of the limited number of orthogonal channels, the spatial reuse of channels results in high level of interferences; this degrades the network performance by lowering its overall throughput. We optimize the network throughput by favoring topologies with well balanced channel reuse. The number of occurrences of a channel q' , denoted by $O_{q'}$, is used to compute the gap between the balanced allocation of channel q and the current allocation.

$$\varphi_q = \text{Max} |O_q - O_{q'}| \quad \forall q, q' \in C \quad \text{Where,}$$

$$O_q = \sum_{j,l \in L} y_{jl}^q \quad \forall q \in C$$

Our aim is then to minimize this gap; this is the second objective function of our model.

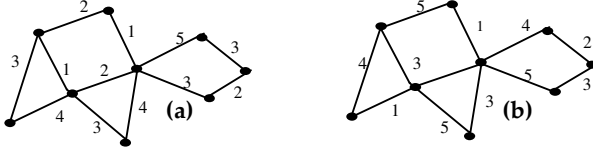


Fig. 2. Same topology with two different channel allocations.

- (a) $O_1=2, O_2=3, O_3=4, O_4=2, O_5=1$ ($\varphi_1=2, \varphi_2=2, \varphi_3=3, \varphi_4=2, \varphi_5=3$),
 (b) $O_1=2, O_2=2, O_3=3, O_4=2, O_5=3$ ($\varphi_1=1, \varphi_2=1, \varphi_3=1, \varphi_4=1, \varphi_5=1$).

$$\text{Min} \sum_{q \in C} \varphi_q . \quad (2)$$

Illustration in Fig. 2 shows that, spatial channel reuse is better in (b) than in (a). The value of $\sum \varphi_q$ in (a) is equal to 12 while $\sum \varphi_q$ in (b) is equal to 5. This is caused by the unbalanced reuse of some channels (i.e. 2 and 3) in (a).

Congested MGs. When all traffic to or from mesh clients (through APs) traverse a subset of network gateways, it may make these gateways congested; this leads to unfair/unbalanced use of gateways (i.e., some gateways are congested while others are barely used). In this paper, we consider fairness, in using gateways, as another performance metric to be optimized.

One of the conflicting objectives we plan to optimize is to minimize this unfair use of MGs, measured by the standard deviation of flows entering network gateways, as given below.

$$\text{Min} \sqrt{\frac{\sum_{l \in G} F_l^2}{\sum_{l \in G} F_l}} . \quad (3)$$

Full Coverage Criterion. The coverage is defined as the size of the physical area where TS has a route to the core network. The area depends on the locations of APs but more importantly on the amount of APs that have a route to the core network. APs have partially overlapping coverage areas as shown in Fig.1. The APs should be located such that all TSs are covered. Constraint (4) is used to make sure that a given TS i is assigned to only one CL j . Inequality (5) implies that a TS i is assigned to an installed and covering mesh node j .

$$\sum_{j \in L} x_{ij} = 1 . \quad \forall i \in I \quad (4)$$

$$x_{ij} \leq a_{ij} t_j . \quad \forall i \in I, \forall j \in L \quad (5)$$

The optimization model is also subject to other constraints given as follows:

$$\sum_{i \in I} d_i x_{ij} + \sum_{l \in L} \sum_{q \in C} (f_{ij}^q - f_{jl}^q) - F_j = 0. \quad \forall j \in L \quad (6)$$

$$\frac{f_{jl}^q}{u_{jl}} \leq y_{jl}^q. \quad \forall q \in C, \forall j, l \in L \quad (7)$$

$$\sum_{i \in I} d_i x_{ij} \leq v_j. \quad \forall j \in L \quad (8)$$

$$F_j \leq M g_j. \quad \forall j \in L \quad (9)$$

$$2y_{jl}^q \leq b_{jl} (z_j^q + z_l^q). \quad \forall q \in C, \forall j, l \in L \quad (10)$$

$$g_j \leq t_j. \quad \forall j \in L \quad (11)$$

$$\sum_{l \in L} y_{jl}^q \leq 1. \quad \forall q \in C, \forall j \in L \quad (12)$$

$$\sum_{q \in C} z_j^q \leq R t_j. \quad \forall j \in L \quad (13)$$

$$x_{ij}, z_j^q, y_{jl}^q, t_j, g_j \in \{0,1\} \quad \forall i \in I, \forall j, l \in L, \forall q \in C \quad (14)$$

$$f_{jl}^q, F_j \in R \quad \forall j, l \in L, \forall q \in C \quad (15)$$

Constraint (6) defines the flow balance for each mesh node j . Constraint (7) stipulates that a flow on a link cannot exceed the traffic capacity of that link. Constraint (8) denotes that the aggregated demand received by mesh node j does not exceed the capacity of the radio access interface. Constraint (9) implies that the flow routed to the Internet is different from zero only when the mesh node installed is a gateway. M is a very large number to limit the capacity of the installed gateway. Constraint (10) forces a link between two CL j and CL l using the same channel q to exist only when the two devices are installed, wirelessly connected and tuned to the same channel q . Constraint (11) ensures that a device can be a gateway only if it is installed. Constraint (12) prevents a mesh node from selecting the same channel more than once to assign it to its interfaces (local channel interference). Constraint (13) with objective function (2) prevents local and global imbalances in channel allocation. Constraint (14) states that the number of links emanating from a node is limited by the number of its radio interfaces. It also states that if a channel is assigned only once to a mesh node, it is a sufficient condition for its existence. We consider the constraint (6) a soft constraint while the remaining constraints are considered hard constraints. The WMN planning system attempts to optimize the three objectives and satisfy all hard and soft constraints as defined above.

3 Solution Approach

The rationale behind our planning is:

- 1) The maximization of the network throughput, by minimizing the level of interferences;
- 2) The minimization of gateways congestion level;
- 3) The minimization of the total deployment cost by selecting a minimum number of routers/gateways and choosing their positions so that the network connectivity is ensured while providing full coverage to all mesh clients.

WMN planning is a fairly complex problem; its difficulty lies in the fact that it tries to simultaneously address all the criteria. Joint optimization of the above criteria is defined as a multi-objective search problem. As stated earlier, solving a Multi-Objective Optimization Problem (MOOP) returns a set of Pareto-optimal solutions. Each solution represents a different trade-off between the objectives that is said to be “non-dominated”. We use a multi-objective approach based on Particle Swarm Optimization (PSO) technique [21] to solve our planning problem.

3.1 Solving Multi Objective Optimization Problem (MOOP)

In the last two decades, there have been growing interests in the field of multi-objective optimization to solve real-world problems. Good introduction to this field of research can be found in [21], [23]. Without loss of generality, we assume that the various objectives are to be minimized. Then, the optimization of a MOP can be formulated as:

$$\begin{aligned} \text{Minimize } y = f(x) &= [f_1(x), f_2(x), \dots, f_N(x)] \\ \text{where } x &= [x_1, x_2, \dots, x_D] \in \text{decision space} \\ \text{and } y &= [y_1, y_2, \dots, y_N] \in \text{objective space} \end{aligned}$$

One of the most difficult parts encountered in practical network design optimizations is the handling of constraints. For a constrained problem, the decision variables x are subject to a set of constraints. Every decision variable vector x in the decision space is evaluated through the objective functions. The objective values are then represented as points in the objective value space (Fig.3).

Definition 1 (Pareto Dominance): For two decision vectors a and b , a is said to dominate b or $a \prec b$ if and only if:

$$\begin{aligned} \forall i \in \{1, \dots, N\} \quad f_i(a) \leq f_i(b) \quad \text{and} \\ \exists i \in \{1, \dots, N\} \quad f_i(a) < f_i(b). \end{aligned}$$

Definition 2 (Pareto Optimality): A decision vector a is said to be Pareto Optimal if and only if a is non-dominated.

Definition 3 (Pareto Front): The Pareto Front is a set of all Pareto Optimal solutions (non-dominated solutions) in the objective value space.

Fig.3 shows that points that lie in the three dimensional area are dominated by the origin point (dotted point) of that area.

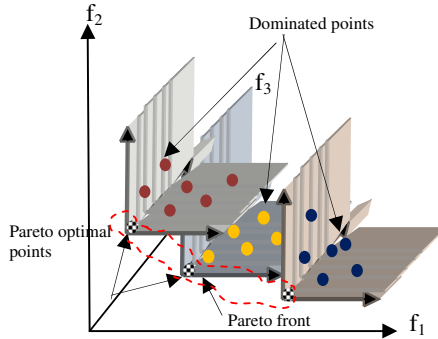


Fig. 3. Pareto Dominance, Optimality and the Front for 3 objective functions

We adapt MOPSO, Multi-Objective Particle Swarm Optimization, as the optimization technique [21], [24] to solve the WMN design problem. We call the variant we use, VMOPSO. Apart from finding the non-dominated solutions, achieving a well-diverse Pareto solution front is the primary goal of the MOOP. We use a crowding distance mechanism in order to maintain diversity of Pareto front solutions and we incorporate a mutation factor (*fmut*) to boost the exploration capability of the standard MOPSO [21]. In the following, we provide more details on how the multi-objective generic model is solved using VMOPSO.

3.2 VMOPSO Algorithm

Further the crowding distance incorporated as the deletion method applied on the external repository *REP* to maintain solutions diversity, we also add a constraint handling mechanism for solving constraints optimization problem, such as WMN design problem.

The crowding distance value, initially defined in [26], is the average distance of two neighboring solutions. The boundary solutions with the lowest or the highest objective function value are given an infinite crowding distance values; thus, they are always selected. This process is done for each objective. The final crowding distance value of a solution is computed by adding the entire individual crowding distance values in each objective value. Personal best solution (*pBest*) and global best solution (*gBest*) are the most important parameters of a particle that the optimizer determines to guide the swarm, in order to obtain a front of optimal solutions. A formal description of VMOPSO is given below.

Algorithm 1: VMOPSO Main Algorithm

Input *fmut*: Mutation factor, *MaxGeneration*
Output *REP*: Repository

- 1: Initialize the swarm (Build feasible solutions that satisfy all the constraints defining the optimization problem)
 - For** each particle *i* in the swarm
 - a. Initialize feasible position,
 - b. Set the personal best guide *pBest* to that position

```

c. Initialize velocity /* see definition below*/
d. Specify lowerBoundi and upperBoundi /*0-1 for integer
   variables*/
e. Set the global best guide gBest to pBest
End For
2: Initialize the iteration counter t=0
3: Evaluate all particles in the swarm /*evaluation of objective functions*/
4: Store non dominated solutions found in step 1 into REP.
5: Repeat
a. Compute the crowding distance values for each j∈REP
b. Sort REP in descending crowding distance values
c. For each particle i in the swarm
   i. Set gBest[i] to the randomly selected particle from the
      top 10% of the sorted REP.
   ii. Compute the new velocity, position of particle i
   iii. Check particle boundaries, if violated change particle
        search direction (i.e., velocity(i) * -1)
   iv. If (t < MaxGeneration*fmut) then mutate
   v. Evaluate particle i
      End for
d. Check for constraints satisfaction
e. Check for non dominance of all particles in the swarm,
   insert non-dominated and feasible solutions into REP and
   delete dominated solution from REP
f. If REP is full then
   i. Compute the crowding distance values for each j∈REP
   ii. Randomly selected particle from the bottom 10% of the
        sorted REP (most crowded portion).
   iii. Replace it with the new solution.
      End if
g. Update pBest
h. Increment t
Until (t= MaxGeneration)

```

During the exploration of the search space, each particle has access to two pieces of information: the best Potential Solution (PS) that it has encountered (*pBest*) and the best PS encountered by its neighbors (*gBest*). This information is used to direct the search by computing velocities: $velocity[i] = iw * velocity[i] + r_1 * (pBest[i] - position[i]) + r_2 * (REP[gBest] - position[i])$, where r_1, r_2 are random numbers in the range of [0,1]. *iw* is the inertia weight. A large inertia value will cause the particles to explore more of the search space, while small one directs the particles to a more refined region. The importance of inertia weight was pointed out by Shi and Eberhart [25] who reported that 0.4 is the best value.

3.3 Solving the WMN Planning Problem Using VMOPSO

Physical Representation Layout. In [16] and [17], authors have shown the benefits of grid topologies over random topologies where coverage, connectivity, average fair capacity, and network throughput are better in grid topologies, especially square grid topologies, than random topologies. In this study, we adopt a *square-grid-like* layout as the physical representation of our WMN planning. Each grid cell corner is a Candidate Location CL where a mesh node can be installed. If a mesh node is

installed at a given CL, it establishes a wireless communication with its eight direct-neighbors. This assumption will increase the chances of selecting a candidate neighbor among the eight with which a wireless link will be set up in the channel assignment procedure with respect to Constraint (13).

Particles Encoding. In Particle Swarm Optimization, a particle, a position in the search space, represents a set of assignments that is a solution to the problem. In our case, a particle is a complex data structure that provides information about user connectivity (x_{ij}), device installation (t_j) and (z_j^q), device connectivity (y_{ji}^q), gateway existence (g_j), link flows (f_{ji}^q), and gateway/backbone link flows (F_j). All decision variables are 0-1 value variables except flow variables (f_{ji}^q, F_j) that are assigned real (float) values. A feasible solution must satisfy all hard and soft constraints. During the search, non-feasible solutions that violate only the soft constraint (6) can be included in the population. This increases the likelihood of a non-feasible solution to mutate and provide a feasible one in later generations. The followings are the phases involved in the resolution of the proposed model.

Building Initial Feasible Solutions. WMN planning problem is a constrained optimization problem; therefore, the initial positions must represent feasible solutions, and thus, need to be designed carefully. Constructing an initial set of feasible solutions that satisfy the constraints (4) to (15) represents the most challenging part in our optimization process.

First, we start by selecting randomly a CL_j from the set of CLs that cover that TS_i (Fig 4.a). An AP (Access Point) is then installed at this location CL_j only if it has not yet been selected. By applying the same procedure to all TSs, we obtain the set S_1 of APs locations that provide full coverage to all TSs. More formally, $S_1 = \{ j \in L, CL_j \text{ covers } TS_i, i \in I \}$. At this stage, constraints (4) and (5) are satisfied and the initial set contains vertices of a disconnected graph as shown in Fig.4.a.

Once the coverage is done, there is a need to augment the set S_1 by adding new MRs (mesh Routers) to connect the APs together. We apply a neighborhood based selection algorithm to find the next node to be inserted. The augmentation algorithm consists, mainly, of choosing the closest neighbor in one component graph to any node of a different component. Then, the path between the two nodes is augmented

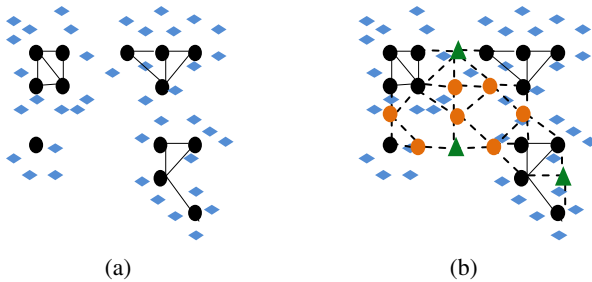


Fig. 4. A Feasible Particle position example. (a): TSs assigned to CLs and a subset S_1 is formed (b): S_1 is augmented and MGs are selected.

(place a router j at the appropriate CL_j $j \in L-S_1$). The algorithm stops when the final graph is connected (see Fig.4.b) and finally, gateways are selected from the set of eligible positions to place gateways. For computational purposes, we use a symmetric adjacency matrix to represent the connectivity graph. We apply the fixed channel assignment algorithm described by Das et al. [27] and we implement Edmonds-Karp's max flow algorithm [28] to assign a value on each link y_{jl} using channel q to route a flow. All remaining constraints (i.e., 6-15) are then satisfied.

Breeding Potential Planning Solutions: The WMN Planning Algorithm. For each particle in the swarm, the iterative algorithm (Algorithm 2) consists of constructing a subset S_1 , mutating it, placing gateways and then assigning flows and channels. The most important phase is the repetitive task of constructing the set S_1 of APs locations to cover all TSs and then mutating it over and over until it satisfies at least all hard constraints. Then, S_1 is augmented to ensure the connectivity.

After this solution-construction process, the velocities, the position and the fitness (values of the three objective functions) of the particles are computed. Then, some of these particles are inserted into the archive provided that they dominate or at least are non-dominated by the previously "archived" non-dominated solutions.

Algorithm 2: Planning Solution

Input $fmut$: Mutation factor, $MaxGeneration$

Output REP : External repository

```

t=0;
Construct_Initial_Soft&Hard_feasible_solutions();
While ( $t < MaxGeneration$ )
  For each particle in the swarm
     $S_1 \leftarrow Mutate(S_1, fmut)$  ;
     $S \leftarrow Augment(S_1)$ ;
     $Y_1 \leftarrow Construct\_connectivity\_matrix()$ ;
     $Y \leftarrow Assign\_channels(Y_1)$ ;
     $G \leftarrow PlaceGateways()$ ;
    Compute_flows();
    Construct_New_Particle() Endfor
  Compute_Velocities();
  Update_Positions();
  Evaluate_Particles();
   $REP \leftarrow Insert\_feasibleNonDominated\_Solutions()$ ;
  Update_ParticuleBest();
   $t++$  ;
Endwhile

```

A position in the search space is a solution to our planning problem; however, the values, returned by Update_Positions() procedure, are not guaranteed to be integers (0 or 1). For this purpose, we add a final process that we call *particle filtering* to allow only particles with a considerable move (to the new position) to change to 0 (respectively 1). If the difference between the two positions (initial and updated one) that a particle gets in the search space goes beyond a given threshold α (based on experiments, we set α to 0.3), then the final position is the reverse of the initial one (i.e., 0 if it was 1 and vice versa); otherwise, the new position is discarded (the

particle remains in its original position). Consequently, all retained positions are then 0-1 integers.

3.4 Complexity Study

Let the number of functions to be optimized be M , and the size of the swarm and the repository be n and N , respectively. In Algorithm 2, the complexity is mainly influenced by checking for feasible non-dominated solutions and the diversity computation operation. However, the cardinality of the set of feasible solutions generated iteratively is much lower than the size of the repository allowed, due to the number of constraints a solution has to satisfy. Consequently, the diversity computation function, based on a crowding factor calculation, is very rarely performed. For checking a particle for its non dominance within $N+n$ particles, $M(N+n)$ comparisons are needed. Therefore, the worst case complexity of this function will be $O(M(n+N)^2)$. Considering the worst case complexity by assuming that the repository truncation is possible, sorting on the basis of each objective will have a complexity of $O(MN\log(N))$. Then, the worst case complexity (with $n+N$ elements in the repository) is $O(M(N+n)\log(N+n))$. Thus, the overall worst case complexity is $O(M(N+n)^2)$.

4 Experimentations and Results Analysis

In this section, we evaluate the performance of our approach. We consider the following key parameters of WMNs: the number of TSSs n , the number of CLs m , the client demands d_i , and the number of radio interfaces R . The purpose of our experimental approach is to study the performance of our model by varying one WMN key-parameter at a time while maintaining others fixed. For this purpose, we define the Standard Setting (SS) of the WMN as the following: SS=[(n :150), (m :49), (d_i :2Mb/s), (u_{ji} :54Mb/s), (v_j :54Mb/s), (M :128Mb/s), (c_j :200), (p_j : $8*c_j$), (R :3), (k :7)]. The algorithm is coded in the Java programming language and all the experiments were carried out on a Pentium M 1.5 GHz.

Unless stated otherwise, we use the standard setting SS. The positions of the n TSSs are randomly generated. A run of our algorithm involves 200 generations each with a population size and archive size of 30 and 20 particles respectively. It must be noted that in our very recent experiments [29], mutating at a rate of 50% of the population leads to the best Pareto front of optimal solutions when compared to optimal solutions. Therefore, we take $fmut=0.5$ as our standard setting for the remaining experiments.

4.1 Performance Evaluation

For each of the following key parameter variation studies (called scenarios), results are reported after 10 runs. Additional filtering process is required to maintain the non-dominance aspect of the collected Pareto fronts. We use OriginPro [30] to plot the 3D objective space graph of planning solutions. For a scaling purpose, the second and third objective values are multiplied by 10^3 . In addition, for each scenario we plot also the devices utilization graphs (only cheapest solutions are considered) in terms of total number of mesh nodes (MN), access points (AP), gateways (MG), and links (Links).

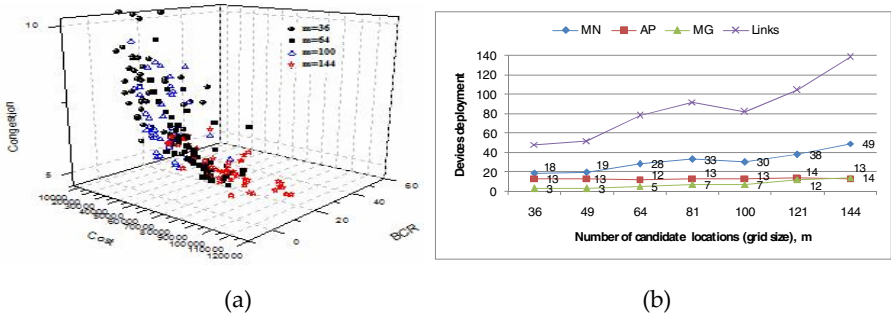


Fig. 5. Effect of changing the number of candidate locations m

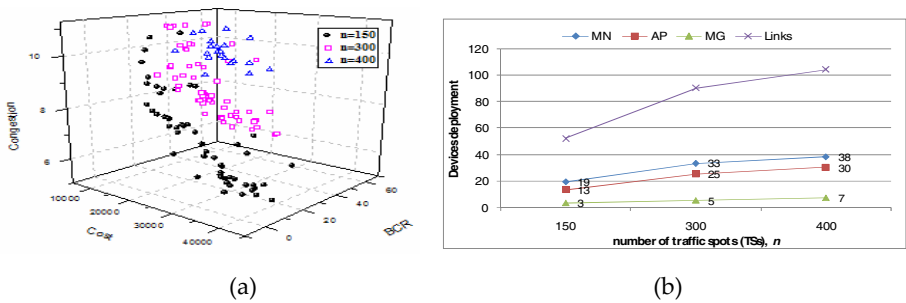


Fig. 6. Effect of changing the number of traffic spots n

Effect of Changing the Number of Candidate Locations m . Notice, from Fig.5.a, that a larger size of grid can improve the network performance (congestion of gateways decreases), but also increases the total deployment cost, which is highly affected by the number of gateways deployed. Therefore, in practice, the network planner has to decide on the appropriate grid size that satisfies both cost and performance requirements. Notice also, that a 10×10 -grid topology is shown to be the best in satisfying the Standard Setting (SS). As shown in Fig.5.b, the number of APs remains relatively stable. A higher number of CLs leads to an increase in the number of routers even for the same number of users. The fact of increasing the number of CLs increases the number of mesh nodes (AP, MR and MG) that are not connected to each other, which leads to install more MRs to construct multi-hop wireless paths between them. On the other hand, the increase of links to ensure connectivity constraints increases the network interference level, as shown in Fig. 5.a (BCR axis).

Effect of Changing the Number of Traffic Spots n . We also study how our algorithm (Algorithm 2) would behave when n varies. Naturally, when n increases (i.e., more mesh clients need to be covered and connected) then more routers need to be connected to the backbone. As shown in Fig. 6.b, the number of APs is increased to cover more mesh clients, in the same time the number of MGs is increased to connect them to the Internet while a smaller increase of MRs is observed to satisfy connectivity constraints. Fig. 6.a shows that the more mesh clients are connected, the

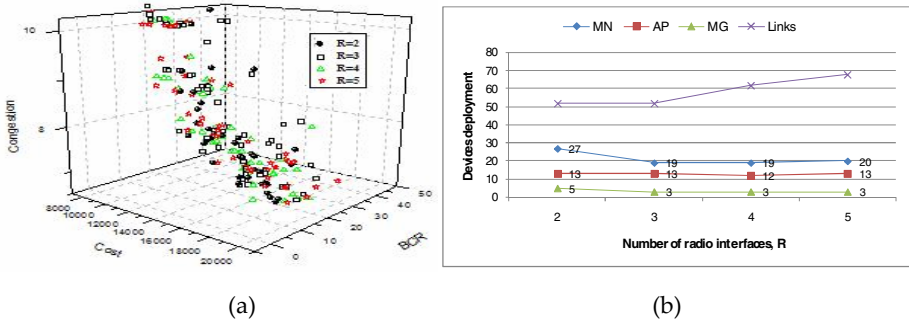


Fig. 7. Impact of the number of radio interfaces on network planning

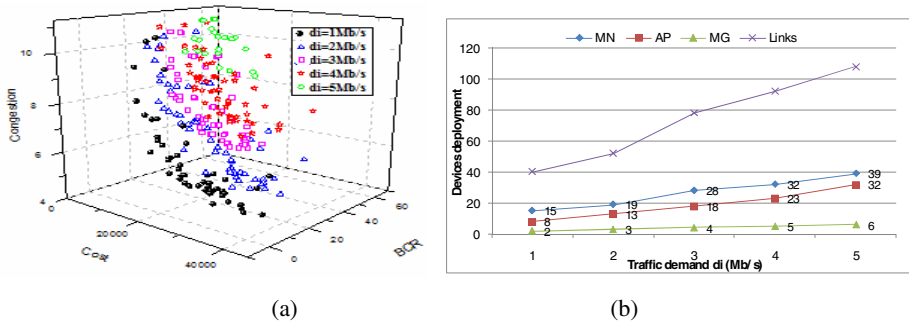


Fig. 8. Effect of varying traffic demand d_i on network planning

more gateways are congested, if the deployment cost is not adjusted accordingly. Given a set of alternative solutions, the network planner makes his/her final decision by selecting the best solution that fits best his/her financial/performance requirements.

Effect of Changing the Number of Radio Interfaces R . We gradually vary the number of radio interfaces from 2 to 5, each with 7 channels. The more radio interfaces we deploy the more links are established, and the less MGs we need. Notice, from Fig. 7.a, that $R=4$ provides the best Pareto front as it has a better solutions spectrum width and cardinality. Also, based on visual observation, the solutions of the same Pareto front ($R=4$) have less MGs congestion level compared to the other Pareto fronts. With respect to devices deployment, Fig.7.b, shows that the cheapest solution returned when $R=4$ is also the cheapest solution amongst all others ($R=2, 3$ and 5) as it requires a smaller number of APs, MGs, and MRs. Regarding the Pareto front, notice that increasing the number of radios increases the width of the spectrum of non-dominated solutions offered to the network planner.

Effect of Varying Traffic Demand. We further study demand variation. The results (Fig.8.a) show that when increasing demand d_i from 1 to 5Mb/s, the number of APs increases. This is expected; new APs are added to guarantee coverage to all mesh clients under capacity constraints. Notice also that the number of MGs increases accordingly to satisfy connectivity constraints by creating new paths to the newly

added APs. However, there hardly exist feasible solutions when d_i is more than 4Mb/s. This is because a 7×7 grid is insufficient to support the $d_i \times n$ demand. Fig. 8.b shows that when d_i increases, the interference level (BCR axis) and gateways congestion level increase for almost all Pareto solutions.

4.2 A Comparison with Related Work

Validating our results against other known models for WMN planning problems turns out to be “impossible” since it is unpractical to compare a set of Pareto (three-dimension) optimal solutions with a one-dimension optimal solution. Moreover, there is no close related work that considers congestion of gateways when designing WMN from scratch. Nevertheless, we can at least check the one common objective function (deployment cost) to see whether the results fall in the same range. We compare our results to the (only) closest related work reported in [20] that considers WMN design from scratch. We refer to the model in [20] as AML and to ours as MOBD. The authors in [20] used the following parameters setup ($d_i=3\text{Mb/s}$, $n=100$, $m=50$, $R=3$ and $k=11$.) and obtained a “single” planning solution which is the mean value over 10 runs. Using the same parameters setup, we obtained 71 non-dominated planning solutions (some of them are shown in Table 2). We report our cheapest and most expensive planning solutions together with the single solution of AML in Table 3.

The solutions of MOBD are numerous and diverse, ranging from very cheap solution (MOBD1 line in Table 3) to very expensive solution (MOBD2 line in Table 3) differing mainly by the measured performance indicators: (1) BCR: interferences over network channels; and (2) S.D: Gateways congestion level.

Results in Table 2 show that our approach tends to provide some solutions which may be more expensive than that of AML in some instances. The network performance is increased by increasing overall network throughput (by minimizing network interferences) and by minimizing network bottlenecks (MG nodes). None of such performance considerations is considered in AML model formulation, which is essentially a single-objective model. This fact led us to compare only the common objective (cost objective function) on a single objective basis. Table 3 shows that MOBD generates from 10% less expensive solution to almost double-price solutions when compared to AML generated solution for the same parameters settings.

Table 2. 9 solutions of MOBD,
 $d_i=3\text{Mb/s}$, $n=100$, $m=50$, $R=3$ and $k=11$

MR	AP	MG	Links	Cost	BCR	S.D.
21	12	3	74	9800	14	10.72
25	15	4	78	12200	13	10.75
27	15	4	80	12600	6	10.90
28	18	4	86	12800	8	10.66
29	16	4	90	13000	15	10.64
21	13	5	70	13200	16	9.88
21	14	5	70	13200	19	9.76
21	14	6	70	15000	6	8.82
34	17	9	124	23000	10	7.25

Only the first 8 solutions and the last (71th) solution are shown in TABLEII.

Table 3. Solutions of MOBD versus the solution of AML

	MR	MG	Links	Cost
AML	23.65	3.3	21.35	10660.0\$
MOBD1	21	3	74	9800.0\$
MOBD2	34	9	57	23000.0\$

MOBD1: Cheapest solution, MOBD2: most expensive solution ($c_f=200\$$, $p_f=8* c_f$).

5 Conclusion

In this paper, we have addressed and formulated the WMN topology design problem. We have considered a simultaneous optimization of deployment cost, network interference level and congestion of gateways while satisfying other criteria. We proposed an efficient nature inspired search algorithm to solve the model formulated whereby different trade-off solutions are provided to the network planner to choose among. We carried out a detailed experimental study, to show and assess the effectiveness of our approach. In the light of the results shown in Section 4, and under many deployment scenarios, we observed the impact of the grid size and the number of radio interfaces on network performance. The variation of the number of mesh clients and the traffic demand has shown how network scalability is handled under our approach. Following the same strand, next we will investigate optimal gateways placement and its impact on network performance.

References

- [1] Benyamina, D., Hafid, A., Gendreau, M., Hallam, N.: Managing WMNs: Analysis and proposals. In: IEEE WIMOB (2007)
- [2] Costa, A.M.: Models and algorithms for two network design problems. Ph.D. thesis, Mtrl (2006)
- [3] Jain, K., Padhye, J., Padmanabhan, V.N., Qiu, L.: Impact of interference on multi-hop wireless network performance. In: ACM MOBICOM (2003)
- [4] Raniwala, A., Chiuch, T.: Architecture and Algorithms for an IEEE 802.11-based Multi-channel WMN. In: IEEE INFOCOM (2005)
- [5] Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: ACM MOBICOM (2004)
- [6] Alicherry, M., et al.: Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio WMNs. In: ACM MOBICOM (2005)
- [7] Sen, S., Raman, B.: Long Distance Wireless Mesh Network Planning: Problem Formulation And Solution. In: 16th international conference on World Wide Web (2007)
- [8] Chen, C., Chekuri, C.: Urban wireless mesh network planning: The case of directional antennas. Technical Report UIUCDCS-R-2007-2874, Dept. of Computer Science, UIUC (2007)
- [9] Chandra, R., Qiu, L., Jain, K., Mahdian, M.: Optimizing the placement of internet taps in wireless neighborhood networks. In: IEEE ICNP (2004)

- [10] Aoun, B., Boutaba, R., Iraqi, Y., Kenward, G.: Gateway Placement Optimization in Wireless Mesh Networks with QoS Constraints. *IEEE J. on Selected Areas in Communications* (2006)
- [11] He, B., Xie, B., Agrawal, D.P.: Optimizing the Internet Gateway Deployment in a WMN. In: *IEEE MASS* (2007)
- [12] Vanhatupa, T., Hannikainen, M., Hamalainen, T.D.: Performance Model for IEEE802.11s Wireless Mesh Networks Deployment Design. *J. Parallel Distrib. Comput.* V 68, 291–305 (2008)
- [13] Max, S., Stibor, L., Hiertz, G.R., Denteneer, D.: IEEE 80211s Mesh Network Deployment Concepts. In: *13th European Wireless Conf.* (2007)
- [14] Robinson, J., Knightly, E.W.: A performance Study of Deployment Factors in Wireless Mesh Networks. In: *IEEE INFOCOM* (2008)
- [15] Li, F., Wang, Y., Li, X.Y.: Gateway Placement for Throughput Optimization in Wireless Mesh Networks. In: *IEEE ICC* (2007)
- [16] Robinson, J., Uysal, M., Swaminathan, R., Knightly, E.: Adding Capacity Points to a Wireless Mesh Network Using Local Search. In: *IEEE INFOCOM* (2008)
- [17] Huang, J.H., Wang, L.C., Chang, C.J.: Throughput-coverage Tradeoff In a Scalable Wireless Mesh Network. *J. Parallel Distrib. Comput.* 68, 278–290 (2008)
- [18] Hsu, C., Wu, J., Wang, S., Hong, C.: Survivable And Delay-guaranteed Backbone Wireless Mesh Network Design. *J. Parallel Distrib. Comput.* 68, 306–320 (2008)
- [19] Beljadid, A., Hafid, A., Gendreau, M.: Optimal Design of Broadband Wireless Mesh Networks. In: *IEEE GLOBECOM* (2007)
- [20] Amaldi, E., Capone, A., Cesana, M., Filippini, I., Malucelli, F.: Optimization Models And Methods For Planning WMNs. *J. Computer Networks* 52(11) (2008)
- [21] Coello, C.A., Lechuga, M.S.: MOPSO: A proposal for multiple-objective particle swarm optimization. In: *IEEE World Cong. on Comp. Intellig.* (2002)
- [22] Kodialam, M., Nandagopal, T.: Characterizing The capacity Region in Multi-radio multi-channel Wireless Mesh network. In: *MOBICOM* (2005)
- [23] Abraham, A., Jain, L.C., Goldberg, R.: *Evolutionary multi-objective theoretical advances and applications.* Springer, Heidelberg (2005)
- [24] Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *IEEE International Conference on Neural Networks* (1995)
- [25] Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: *IEEE International Conference of Evolutionary Computation* (1998)
- [26] Raquel, C.R., Naval, C.: An effective use crowding distance in multi-objective optimization. In: *ACM Conf. on Genetic and Evol. Comp.* (2005)
- [27] Das, A.K., Alazemi, H., Vijaykumar, R., Roy, S.: Optimization Models for Fixed Channel Assignment in Wireless Mesh Networks with Multiple Radios. In: *IEEE SECON* (2005)
- [28] Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM* (1972)
- [29] Benyamina, D., Hafid, A., Gendreau, M.: Wireless Mesh Networks Planning: A Multi-objective Optimization Approach. In: *IEEE BROADNET* (2008)
- [30] <http://www.originlab.com>

Novel Analytical Delay Model and Burst Assembly Scheme for Wireless Mesh and Optical Burst Switching Convergence

Jihene Rezgui, Abdeltouab Belbekkouche, and Abdelhakim Hafid

Network Research Laboratory
University of Montreal, Canada
{rezguiji, belbekka, ahafid}@iro.umontreal.ca

Abstract. Wireless Mesh Networks (WMN) have attracted increasing attention from the research community as a high-performance and low-cost solution to last-mile broadband Internet access. In the other side, Optical Burst Switching (OBS) is a promising access technology that uses optical fiber with burst switching paradigm. In this paper, we propose a novel Metropolitan Area Network (MAN) architecture, called Optical Burst Wireless Mesh Architecture (OBWMA) which integrates WMN at the user access side and OBS at the core of the MAN. OBWMA aims to combine advantages of both WMNs and OBS networks, such as large coverage at low cost and bandwidth availability. We specify the details of the interconnection and the internetworking of WMNs and the OBS network in OBWMA. Moreover, we develop an analytical model to compute the end-to-end delay in OBWMA in order to support flow requests with delay constraints. Furthermore, we propose a Control Bridge (CB) that ensures Quality of Service (QoS) mapping at the border between the WMN and the OBS parts. Also, we propose a burst assembly scheme, called Adaptive Hybrid Burst Assembly scheme (AHBA). Simulation results using ns-2 demonstrate the feasibility of OBWMA and the validity of our analytical model.

Keywords: MAN, WMN, OBS, QoS, DiffServ, Wireless Optical Convergence.

1 Introduction

Metropolitan Area Networks (MANs) are public networks aimed to interconnect high-speed core networks (Wide Area Networks) and relatively low-speed access networks (Access and Local Area Networks). In the context of Next Generation Networks (NGNs), MANs are required to offer the following characteristics [1]: dynamic bandwidth provisioning, scalability, upgradability, efficient and flexible use of resources, Quality of Service (QoS) differentiation, reliability, high throughputs and short delays. Meanwhile, end-users are becoming more and more bandwidth hungry because of data, voice and multimedia applications that have grown exponentially over the past several years; these applications are expected to continue growing over the next years. Thus, a MAN architecture which connects the end-user

to the Internet, while supporting the increasing bandwidth demand and satisfying NGNs specifications, is needed today. In this paper, we study the integration of WMNs and OBS networks in a MAN architecture that takes into consideration the above requirements.

Wireless mesh networks (WMNs) have recently emerged as a promising technology for the next-generation wireless networks. A WMN consists of two types of nodes: Mesh Clients (MCs) and Mesh Routers (MRs). The MRs form an infrastructure which forwards the traffic between MCs and the Internet. In general, MRs have minimal mobility and operate just like a network of fixed routers, except being connected by wireless links through wireless technologies such as IEEE 802.11. A subset of the MRs is connected to the Internet (via wired or wireless links, e.g., 802.16 links); they are called gateways. A WMN has numerous benefits, such as the reduction of installation costs because only a few MRs may have cabled connections to the wired network [2]. Moreover, it has a large-scale deployment since it is a multi-hop network that offers long distance communications through intermediate nodes. Another considerable advantage is the reliability due to the redundant paths between each pair of nodes in the network.

WMNs have been widely deployed to deliver wireless services for a large variety of applications, such as broadband home networking, community and neighborhood networking, metropolitan area networks, health and medical systems, etc. Some of the WMN features are to support ad hoc networking: capability of self-forming, self-healing, and self-organization. Also, a WMN allows multiple types of network access, compatibility and interoperability with existing wireless networks and multi-channel multi-radio operation. This last feature is very important because it increases the network capacity. Indeed, IEEE 802.11 offers multiple non-overlapping channels (e.g., 3 and 12 channels for 802.11b and 802.11a, respectively). Each node could be equipped with multiple radios which increases significantly the throughput. Moreover, the integration of WMNs with other networks, such as the Internet, cellular, IEEE 802.16 and sensor networks, can be accomplished through the gateway and bridging functions in the MRs.

In the other side, Wavelength Division Multiplexing (WDM) is an attractive technology to support the huge amount of bandwidth required by the core of the MAN network. It uses the potential capacity in optical fibers that contains many wavelengths able to carry (potentially) tens of Tbps using statistical multiplexing. This potential requires good switching technology to efficiently exploit it. OBS (Optical Burst Switching) [3] is a good switching paradigm candidate to fill this need. It has received an increasing interest from researchers over the last several years since it presents a good tradeoff between traditional Optical Circuit Switching (OCS) and Optical Packet Switching (OPS). OCS is relatively easy to implement but suffers from poor bandwidth utilization and coarse granularity; OPS has a good bandwidth utilization and fine granularity but suffers from complex implementations because of the immaturity of the current technologies, such as optical buffers and ultra fast optical switches [3].

In OBS networks, data packets with the same destination are aggregated in bursts of variable lengths at the ingress node; this process is called Burst Assembly. After burst assembly, a Control Packet (also, called Burst Header Packet) is sent, using a dedicated control wavelength, from source to destination in order to reserve the

required resources along a lightpath. This control packet is subject to Optical-Electric-Optical (OEO) conversions at each core node (OBS switch) where it receives an appropriate processing to make resource reservation for its data burst. After a delay called Offset Time (OT), the corresponding data burst is sent, on one of the data wavelengths, through the same lightpath without any buffering requirement inside the OBS network. The huge bandwidth and the high flexibility of OBS, added to its simplicity of implementation (using the existing infrastructure), efficient utilization of resources, QoS and differentiated services support, make OBS an excellent candidate to play the role of a core network in a next generation MAN.

Therefore, the integration of WMNs and OBS networks in a novel architecture is an interesting idea to explore. In this paper, we propose a novel MAN architecture, called Optical Burst Wireless Mesh Architecture (OBWMA), which is composed of WMNs at end-user access part and an OBS network at the core part of the MAN. Whereas WMNs offer coverage to the end-users, the OBS network connects several WMNs to the Internet using its huge bandwidth capacity. The operation of OBWMA and the internetworking between WMN and OBS is based on the Internet Protocol (IP) which is adopted as the basis of the NGNs. Our contributions in this paper can be summarized as follows: (a) A novel MAN architecture (OBWMA) integrating, for the first time, multi-channel multi-radio WMNs and OBS networks; (b) An analytical model to compute end-to-end delay in OBWMA; (c) A Control Bridge (CB) which ensures QoS mapping between WMN and OBS; and (d) A novel Adaptive Hybrid Burst Assembly scheme (AHBA).

The paper is organized as follows: Section 2 presents related work. Section 3 describes OBWMA architecture. In Section 4, an analytical model to compute the end-to-end delay inside OBWMA is presented. In Section 5, we present the proposed control bridge and burst assembly scheme. Section 6 presents simulation and analytical results. Finally, Section 7 concludes the paper.

2 Related Work

The concept of wireless and wired convergence is becoming more and more attractive in both academia and industry communities. This trend for reducing the gap between the wireless and the wired domains is motivated by the adoption of NGNs as a framework solution for the next generation Internet that recommends this kind of convergence. In [4], Fixed Mobile Convergence technology (FMC) is proposed. FMC provides seamless services via a combination of fixed (optical)/wireless broadband and wireless access network technologies. The authors in [5] proposed integrated optical and wireless services in the access network; they defined several optical wireless integration scenarios. The integrated network performance has been evaluated through simulations. Their results demonstrate that optical wireless integration decreases access point complexity, increases the capacity of wireless networks and promotes mobility in access networks. Decreasing access point complexity is achieved by integrating the optical access system (Optical Line Terminals (OLTs) of Passive Optical Network (PON)) and wireless base stations at the edge node. The authors in [6] studied a system that integrates GEneralized Passive Optical Network (GEAPON) OLTs and WiMAX base stations at the edge nodes; the proposed system extends the WiMAX antenna using the GEAPON optical link and the Optical Network Unit (ONU). The ONU

aggregates the incoming requests from the WiMAX subscriber stations and sends them towards the edge node. When the requests reach the edge node, the OLT interacts with the WiMAX base station to allocate the necessary bandwidth so that the subscribers get the required QoS when their traffic passes through the WiMAX and PON networks. The authors in [7] proposed an integrated network architecture composed of GEAPON and WiMAX (802.16d/e) to reduce the capital and operational expenditure (CapEx and OpEx). To validate their architecture, they studied the QoS support and the wireless network throughput.

We note that Passive Optical Network (PON) is often used in the context of wireless and optical convergence (from the optical side). However, PON uses Time Division Multiplexing (TDM) and often tree topology. Hence, we believe it is not appropriate to use PON as a core network; instead, it is more appropriate to use PON as an access network in the context of Fiber-To-The-Home (FTTH) paradigm.

The authors in [8] proposed a simple paradigm based on connecting WLAN and OBS networks. They investigated the use of the two wireless access mechanisms: DCF and PCF; and different packet sizes without more details. In [9], the authors performed a simulation based study of TCP performance over an architecture composed of OBS at the core network and 802.11 at the access network. We believe that the contributions in [8] and [9] are in the right direction since they show the interest of interconnecting 802.11 and OBS technologies. However, they remain in the stage of *proof of the concept* without exploring the interconnection concept deeply and without a global view of the performance of the converged network (the wireless and the optical parts). Moreover, to the best of our knowledge, this is the first time that the integration of multi-channel multi-radio WMNs and OBS networks is studied.

3 The Proposed Architecture

Fig. 1 shows the structure of the proposed architecture where a number of WMNs (of medium size) are interconnected between them and connected to the Internet through a core OBS network. In fact, in this architecture, the gateways (one or more) of each WMN are connected to an OBS edge node. For a given OBS edge node, the WMNs connected to it are called *home WMNs* while the other WMNs (in OBWMA) are called *foreign WMNs*. The connection between an OBS edge node and their home WMNs could be performed using a dedicated device or simply using a wired connection. This point is discussed in subsection C.

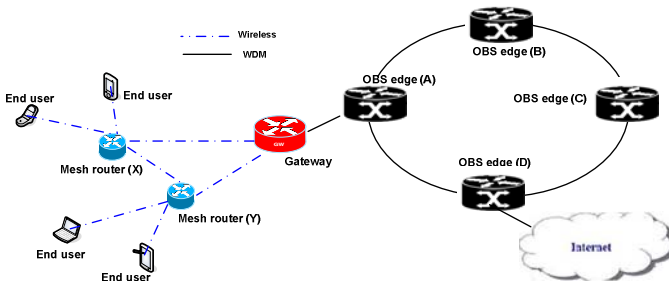


Fig. 1. Overview of OBWMA

3.1 The WMN Part

The first key part of OBWMA is the WMN part (see Fig. 2) where the MRs (e.g. MR_3, MR_{17}) aggregate and forward the traffic from their MCs to the gateways. The MRs communicate with each other to form a multi-hop wireless network. This wireless network forwards the user traffic to the gateways (e.g. P_1 and P_2 in Fig. 2) which are connected to the Internet and other WMNs in OBWMA via the OBS core network.

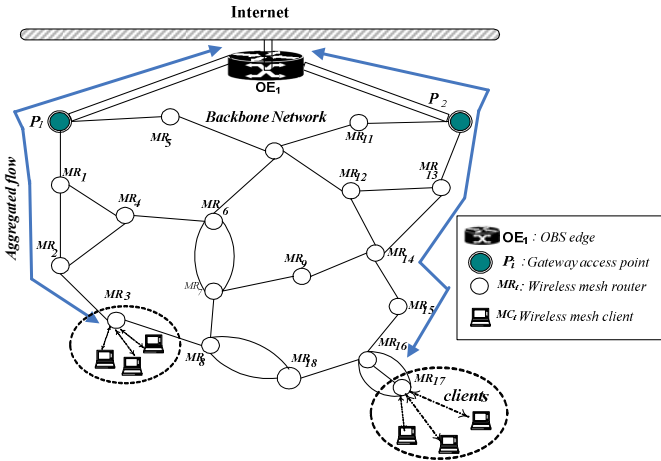


Fig. 2. A schematic sketch for WMN over OBS

A WMN allows mesh nodes or gateways to communicate with each other without being routed through a central switch point, eliminating centralized failure and providing self-healing and self-organization. Although decisions on traffic are made locally, the network can be managed globally. Furthermore, since each MR may aggregate traffic flows for a large number of mobile MCs, the aggregate traffic load of each MR changes very rarely. In WMN infrastructure, some MRs are also equipped with a gateway capability through which they interface with the wired network (OBS core network in our case). In such networks, traffic is mainly routed by the WMN wireless backbone between the MCs and the Internet through the gateways. In addition, the traffic distribution is typically skewed as most of the user traffic is directed to/from the wired network.

Providing QoS in the wireless part depends upon how well the network capacity is estimated. This estimation is difficult to obtain because, compared to wired networks, the links in WMNs are inherently shared (because of interferences) and difficult to isolate. This fact makes the performance of WMNs difficult to control. It is crucial to control the traffic to guarantee the QoS requirements (e.g., end-to-end delay) [10]. The reason is that interferences among links cause performance degradation, e.g., two interfering links that are active simultaneously often provide much less throughput than two separated links. To guarantee QoS, OBWMA accepts a new flow request only when the required flow end-to-end delay in WMN part is satisfied (see section 5).

Architecture and routing protocols mechanisms of WMNs have been extensively studied in literature [2]. In OBWMA, we use DSDV [11] as the routing protocol of the wireless Part. It is worth noting that any other routing protocol for WMNs can be used.

3.2 The OBS Part

The OBS core network of OBWMA is a ring network composed of a set of OBS edge/core nodes. In fact, each OBS node in the core network can receive and send the traffic of its home WMNs (which is the role of an OBS edge node) and route the transit traffic of foreign WMNs (which is the role of an OBS core node). The ring topology is widely used in MANs because of its characteristics of failure recovery and scalability. However, other kinds of topologies, such as mesh and regular topologies could be used for OBWMA.

Burst assembly is a key mechanism in the OBS network where incoming packets from the client networks with the same destination are aggregated in data bursts according to some criteria, such as QoS (e.g., delay). There exist mainly three schemes of burst assembly [12]: (a) time-based schemes that use a maximum time threshold before forming the burst; (b) size-based schemes that use a maximum burst size threshold before forming the burst; and (c) hybrid schemes that use both time and size thresholds. For OBWMA, we propose an adaptive hybrid scheme, called Adaptive Hybrid Burst Assembly scheme (AHBA) which uses adaptive time and size thresholds to allow QoS mapping between WMN and OBS, i.e., the translation of QoS constraints from the wireless domain (represented by the WMN) to the optical domain (represented by the OBS network). AHBA is discussed in Section 5.B.

In the OBS network, resource reservation has an end-to-end scope and it is performed by control packets which are sent on dedicated control wavelength(s). Whereas, generally, in OBS networks one-way reservation is adopted to minimize the end-to-end delay, two-way reservation allows preventing burst losses inside the OBS network at the cost of an increase of the end-to-end delay. For OBWMA, we adopt the one-way resource reservation scheme to alleviate the increase in end-to-end delay.

Wavelength assignment has an important impact on OBS networks, especially, when wavelength converters are sparse or not used at all, at the core nodes. In fact, wavelength converters are expensive and have not yet reached their technological maturity. Thus, we do not use wavelength converters in the OBS core network.

In OBS networks, Shortest Path routing (SP) is often used since it ensures optimal resource utilization. However, adaptive and multipath routing could be used to improve the burst loss rate performance. Nevertheless, adaptive and multipath routing approaches suffer from issues, such as routing path loops, out-of-order delivery and jitter [13]. Hence, we adopt SP routing for the OBS core network of OBWMA.

3.3 WMN and OBS Interconnection and Internetworking

The interconnection of WMNs and the OBS core network in OBWMA is ensured by simple wired connections between the gateways of WMNs and the OBS edge nodes.

This interconnection is performed in the electronic domain. It is worth noting that the storage capability of the electronic domain makes it omnipresent in WMN gateways and OBS edge nodes. However, this interconnection has to be carefully provisioned with bandwidth in order to prevent this part of the network from forming a bottleneck. Since narrowing the gap between wireless and optical worlds is one of the objectives of NGNs, we can use optical fibers for the wired connection. Nevertheless, electronic domain will always be present in gateways and OBS edge nodes because of the lack of optical buffers in one hand, and the advanced buffering technology of electronic domain in the other hand. Besides, a sophisticated device which incorporates a gateway with an OBS edge node in a single device could be conceived. Hence, this device will contain wireless, electronic and optical compartments. For economical and simplicity of realization reasons, we propose to use, exclusively, copper-based wired connections between gateways and OBS edge nodes in OBWMA. In fact, copper-based connections are cost effective and could be used with the existing WMN and OBS equipments, namely, gateways and OBS edge nodes.

The internetworking between WMN and OBS is based on the Internet Protocol (IP) which is designated to be the cornerstone of the operation of next generation networks. Fig. 3 shows the protocol stack of OBWMA and Fig. 4 shows the packet flow inside it. In Fig. 3, we can see that the protocol stack of OBWMA is composed of the protocol stacks of WMN and OBS. Specifically, 802.11 protocol is responsible for the operation of the WMN part MAC layer while the OBS protocol is responsible for the operation of the OBS part.

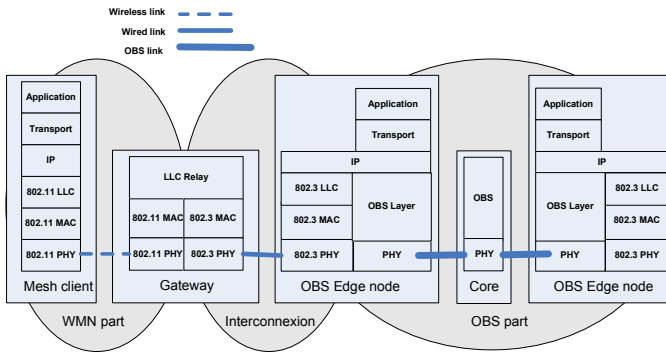


Fig. 3. The protocol stack of OBWMA

The interconnection between WMN and OBS parts is ensured by the 802.3 protocol. Hence, when the user data traverses OBWMA from end to end, it passes through the WMN part MAC layer as a 802.11 frame. Then, at the interconnection MAC layer it becomes a 802.3 frame. Finally, it passes the OBS part layer two as a data burst, before becoming again a 802.3 frame at the MAC layer of the OBS egress node (see Fig. 4).

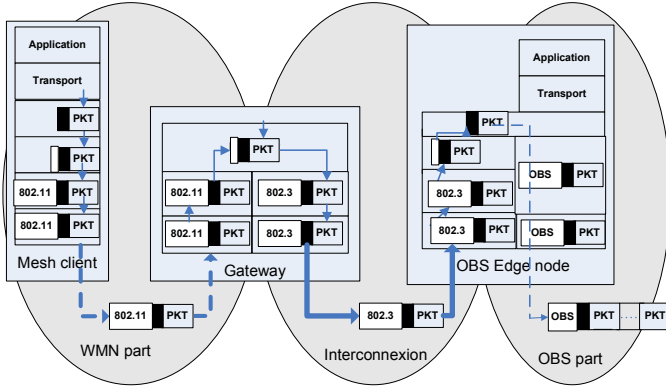


Fig. 4. Packet flow through the WMN and the OBS parts

4 End-to-End Delay Performance Model

As a next generation MAN network, OBWMA has to provide QoS guarantee capability. End-to-End delay is an important QoS constraint, especially, for delay-sensitive traffic. To deal with this constraint, we propose an analytical model to compute end-to-end delay in OBWMA. We characterize the traffic as Poisson process with mean packet arrival rate λ_i packets/s for each node i in the network (WMN or OBS node). It is well known that the combination of two or more Poisson traffics is Poisson traffic. For example, if a Poisson traffic A with mean packet arrival rate λ_A is combined with a Poisson traffic B with mean packet arrival rate λ_B the resulting traffic is still Poisson but with mean packet arrival rate $\lambda_A + \lambda_B$. Hence, if we suppose that the end-users traffic is Poisson, the incoming traffic in intermediate wireless MRs, gateways and OBS edge nodes is always Poisson. Furthermore, we assume that the packet size in WMNs is exponentially distributed with mean L and the maximum burst size in the OBS part is B .

4.1 End-to-End Delay in WMN

Before transmitting a packet, each node in a WMN counts a random timer which is exponentially distributed with mean Backoff duration $\frac{1}{\xi}$. The average service time of a mesh router MR_i , noted b_i , using channel $k \in \{1..NC\}$ is expressed as follows:

$$b_i = \frac{\frac{1}{\xi} + L}{1 - INTER_i} \tag{1}$$

where θ_k is the bandwidth capacity of k^{th} channel and $INTER_i$ is the interference ratio representing interferences between MR_i and its neighboring Mesh Routers (MRs).

In the case of no interferences,

$$INTER_i = 0, \quad (2)$$

If interferences exist, we consider that all interfering MRs have the same probability to access the medium. In this case:

$$INTER_i = \frac{L}{\theta_k} \times \left(\sum_{j \in N_{ct}} \lambda_j \right), \quad (3)$$

where λ_j is the packet arrival rate in a MR_j , and N_{ct} is the set of MRs that can contend for channel k . It is worth noting that $INTER_i$ is always in the range $[0, 1]$, since when it reaches value 1, the average service time tends to infinity; in this case, the majority of packets are dropped due to the high level of interferences.

The end-to-end delay for each path in the WMN part is determined by computing delay at each intermediate MR as follows:

$$d_{WMN} = \sum_{i \in PATH} b_i \quad (4)$$

where $PATH$ is the set of the nodes in the end-to-end path from the MC to the gateway.

4.2 End-to-End Delay in OBS

The delay in the OBS part is mainly composed of the assembly delay (at the edge node), noted d_a , and the offset time delay, noted d_o . The offset time d_o is the delay that separates the transmission of the control packet and the transmission of its data burst. This delay is useful to compensate the processing and the queuing time of the control packet at each intermediate node. Otherwise, the data burst could reach and surpass its control packet, in which case, it will be dropped since no resources have been reserved for it at this part of the OBS network. If AHBA scheme is used (see Section 5.B), a burst is formed if its length reaches size B or if the maximum assembly time T_a is reached. Hence, the average number of packets in a burst at an OBS edge node e is:

$$N = \min\left(\left\lfloor \frac{B}{L} \right\rfloor, \lfloor \lambda_e T_a \rfloor\right) \quad (5)$$

The WMN packet number p in the assembly process at the OBS edge node e , will undergo delay T_p in average, where T_p is given by:

$$T_p = \frac{(N-1) - (p-1)}{\lambda_e} = \frac{N}{\lambda_e} - \frac{p}{\lambda_e} \quad (6)$$

and the average assembly delay is:

$$d_a = \frac{1}{N} \sum_{p=1}^N T_p = \frac{N}{\lambda_e} - \frac{1}{N\lambda_e} \sum_{p=1}^N p = \frac{1}{2\lambda_e} (N-1) \quad (7)$$

Therefore, the mean OBS delay is:

$$d_{OBS} = d_a + d_o \tag{8}$$

where d_o is the sum of control packet queuing and processing times (at each intermediate node). Hence, we can estimate d_o if we can estimate the mean number of hops in the OBS part of OBWMA. Since the traffic in WMN is, generally, destined to/from the Internet, we can estimate the mean number of hops in the OBS core network. This is performed by computing the mean number of hops between the OBS edge node (or nodes) connected to the Internet and the other OBS edge nodes.

The end-to-end delay in OBWMA (i.e., the delay that a packet undergoes from the MC in the WMN part to the OBS egress node connected to the Internet) is calculated as follows:

$$d_{OBWMA} = d_{WMN} + d_{OBS} \tag{9}$$

5 Quality of Service Provisioning

Quality of service (QoS) provisioning is a mandatory functionality for NGNs. For this reason, we propose a QoS provisioning mechanism for OBWMA. This mechanism operates at the border between the OBS and WMN parts of the network. It is based on service differentiation. For service differentiation, and without loss of generality, we consider two classes of traffic: (a) Class of service 1 with quality of service requirements (e.g., delay); and (b) class of service 2 with no QoS requirements (best-effort service). We suppose that user flow requests (at MCs) come with their classes of service (class-1 or class-2) and their maximum end-to-end delays (Δ_{delay}). In addition, the flow request packet contains a field for accumulated delay in the WMN part d_{WMN} . d_{WMN} is updated at each intermediate WMN node. After the flow request is accepted, the 802.11 frames in the WMN part, the 802.3 frames in the interconnection part and the control packets in the OBS part contain a field for the class of service. This field takes value 1 for class-1 and value 2 for class-2. Fig. 5 shows the main fields of the flow request packet.

Source	Destination	Class-i	Δ_{Delay}	d_{WMN}
--------	-------------	---------	------------------	-----------

Fig. 5. Main fields of the flow request packet

5.1 The Control Bridge

In OBWMA, flow requests arrive from the WMN part to the OBS core network with end-to-end delay QoS requirements. In this paper, we consider only end-to-end delay, however, other QoS constraints could be considered (e.g., loss rate). Whenever the OBS edge node receives a flow request, it checks its class of service. Flows of class-1 should have to be accommodated with firm guarantees of QoS constraints but flows

of class-2 could be accommodated with the available resources and without guarantees of QoS constraints. To do so, we propose a Control Bridge (CB) which ensures QoS (delay) mapping between the two parts of the network. Fig.6 depicts the architecture of the proposed CB. It has two interfaces connecting the OBS edge node and the 802.11 gateway. The CB is located at the OBS edge node and has a global view of the state of burst assembly buffers.

For QoS mapping, we propose a novel burst assembly scheme (AHBA) with the main objective of realizing the mapping of delay constraints between WMN and OBS.

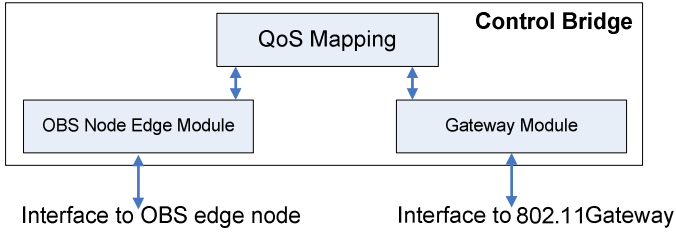


Fig. 6. Control Bridge Architecture

5.2 Adaptive Hybrid Burst Assembly Scheme

For the burst assembly process, we suppose that each OBS edge node has one buffer for each other OBS edge node destination and each class of traffic. Hence, each assembly buffer is identified by (destination, class of service) where destination is the destination OBS edge node and class can take values 1 or 2. Adaptive Hybrid Burst Assembly (AHBA) is a burst assembly scheme which takes into consideration two parameters: (1) Maximum Assembly Time (MAT); and (2) Maximum Burst Size (MBS). In addition, using AHBA, the CB has the ability to act directly on the assembly buffers by tuning or fixing their parameters (e.g., MAT). Indeed, parameters MAT and MBS for traffic of class-2 are fixed. This is done by fixing the burst size to a suitable value (e.g., 10KB) and then fixing assembly time to a reasonable value. The assembly time has to prevent excessive waiting time for class-2 traffic packets in the assembly buffer when the arrival rate of these packets is very low. Generally, this will result in fixed (class-2) bursts size inside the OBS network, which is a suitable property for OBS networks performance [14]. For class-1 traffic, the CB could tune the maximum assembly time of a class-1 assembly buffer to meet the quality of service requirement of a flow request in terms of end-to-end delay. Hence, upon the receipt of a class-1 flow request, the CB checks whether the corresponding assembly buffer could satisfy its end-to-end delay constraint. If it is the case, the flow request could be accommodated; otherwise, the CB tries the possibility of tuning the assembly time of the corresponding class-1 traffic assembly buffer in order to meet the delay requirement of the flow request. Noting that a buffer assembly time could be decreased for a given flow request but it could not be increased again only after the end of this flow. To compute the delay of the flow request, the CB extracts the required end-to-end delay noted Δ_{delay} and the WMN delay d_{WMN} from the flow

request packet. Then, the CB computes the OBS delay d_{OBS} using Eq. 8. The sum of d_{WMN} and d_{OBS} must be less or equal Δ_{delay} :

$$d_{WMN} + d_{OBS} \leq \Delta_{delay} \tag{10}$$

Otherwise, the CB tries to decrease the assembly time of the corresponding buffer by decreasing the maximum assembly time T_a . However, excessively decreasing T_a could result in forming data bursts of size near to that of an IP packet. This could eliminate the advantage of statistical multiplexing of the OBS core network and degrades its performance in terms of resource utilization. To tackle this issue, we introduce a new parameter, called MINimum Burst Size (MINBS), which is used to guarantee a minimum burst size. MINBS could be expressed as a multiple of an IP packet size L , e. g.: $MINBS = 3 \times L$, which means that the minimum number of IP packets in a burst is 3. Thus, Eq. 5 becomes:

$$N = Max(Min(\lfloor \frac{B}{L} \rfloor, \lfloor \lambda_e T_a \rfloor), MINBS) \tag{11}$$

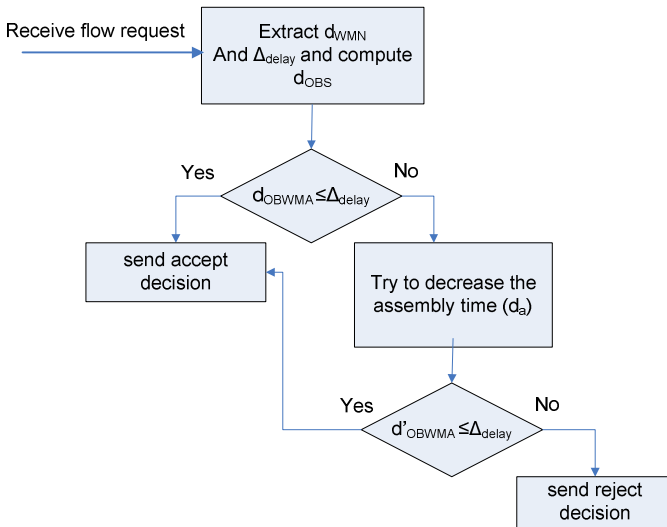


Fig. 7. The operation of AHBA

If a flow request delay requirement could not be met by decreasing the assembly delay d_a , this request is simply rejected, and a reject message is sent back to its source node in the WMN part. Fig. 7 shows the operation of AHBA.

It is worth noting that decreasing burst assembly time for a given flow and increasing it again at the end of this flow could increase the jitter of other flows in the network. However, this variation in the jitter is controllable and could be calculated; it is simply the difference between the assembly times before and after decreasing or

increasing the maximum assembly time T_a . In addition, the value of the jitter could be sent to the corresponding destination OBS node, each time the maximum assembly time is increased or decreased. However, this is out of the scope of this paper.

6 Numerical Results

In this section, we conduct simulations using ns-2 simulator [15] and present numerical results of the proposed delay analytical model to evaluate the performance of OBMWA. Our goal is to: (a) validate the operation of OBMWA, especially, the interconnection of the WMN part and the OBS part using ns-2 simulator; (b) evaluate the proposed analytical end-to-end delay model and compare it to simulation results of the end-to-end delay; and (c) measure the impact of varying the maximum assembly time, the IP packet size in the WMN part and the burst size in the OBS part. We consider only end-to-end delay as the main metric in OBMWA.

We use the topology illustrated in Fig. 8 to perform simulations where real-time traffic flows arrive at each wireless MR according to Poisson process. The traffic load expressed in the figures is the ratio [utilized bandwidth/ bandwidth capacity (with no traffic in the network)] at MR 1.

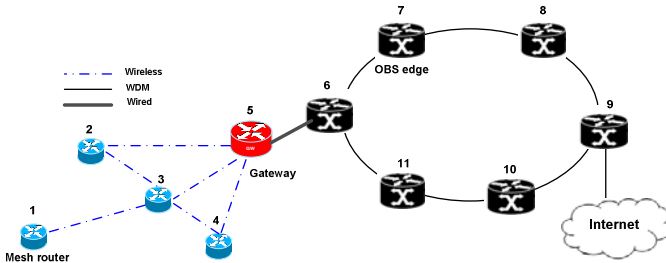


Fig. 8. Simulations topology

For the WMN part, the radio transmission range r takes one of the following values: 150 meters, 200 meters and 250 meters and the transmission interference R of each wireless station is 550 meters. Also, we fix the WMN packet size to 1000 KB unless stated otherwise.

For the OBS part, we assume that each single fiber link is bidirectional and has the same number of wavelengths. We fix the number of wavelengths to 12 wavelengths per fiber link. A larger number of wavelengths will have no impact on the presented results since it will increase the capacity of the OBS core network. Each OBS node can receive and route traffic. That means that each node in the OBS core network plays the roles of both edge node and core node at the same time. Moreover, unless stated otherwise, we fix burst assembly parameters as follows: (a) Maximum Assembly Time (MAT) to 0.1 s; (b) Maximum Burst Size (MBS) to 10000 KB. Also, we use Last Available Unused Channel with Void Filling (LAUC-VF) [16] algorithm for wavelength assignment in OBS edge nodes.

Fig. 9 shows the mean end-to-end delay for OBWMA using the analytical model and simulations. The results demonstrate clearly that the proposed model is quite accurate. In fact, while the mean end-to-end delay (over all of the loads) using the analytical model is 0.077, the mean end-to-end delay using simulations is 0.083; the standard deviation (over all of the loads) between the two is less than 2%. Moreover, Fig. 9 shows delay curves for WMN part and OBS part to show the contribution of each one of them to the overall end-to-end delay in OBWMA. We observe that OBS has the highest contribution to the end-to-end delay, especially, at very low loads. This is explained by the fact that burst assembly takes more time at low loads which affects the overall end-to-end delay.

Fig. 10 shows the end-to-end delay when varying MAT from 0.1 to 0.3 s. The results show that the end-to-end delay decreases whenever the MAT decreases.

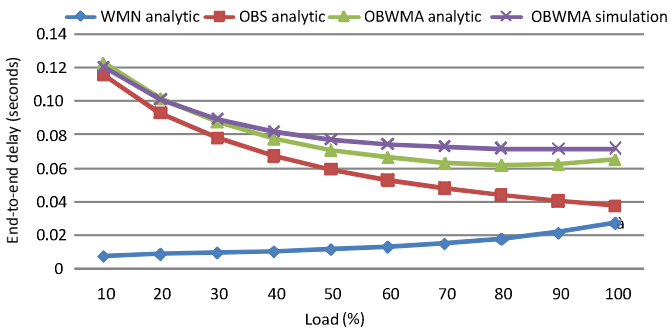


Fig. 9. End-to-End delay: analytic vs. simulations

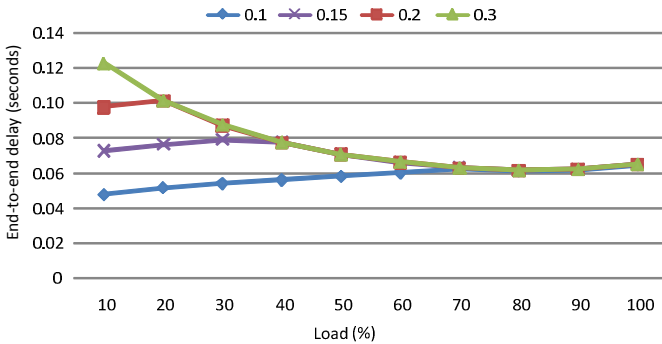


Fig. 10. Impact of maximum burst assembly time (seconds) on end-to-end delay

Fig. 11 shows the impact of varying WMN packet size on the delay. We consider values 500, 1000 and 1500 KB. We observe that the more the packet size increases, the more the delay decreases. This can be explained by the fact that at the same traffic load, when packet size is bigger, the number of packets in the WMN part is reduced and collisions are less likely to occur. Hence, the number of retransmissions due to collisions and, consequently, the delay are reduced.

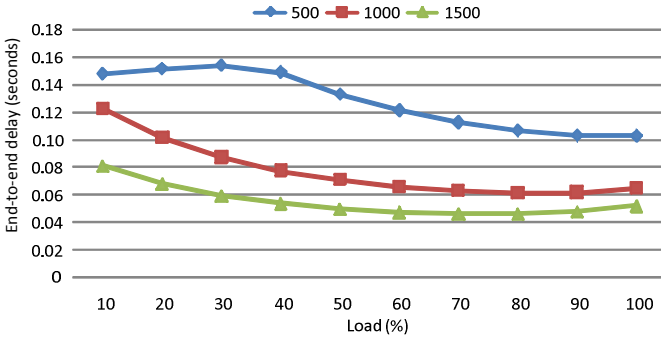


Fig. 11. Impact of WMN packet size (KB) on end-to-end delay

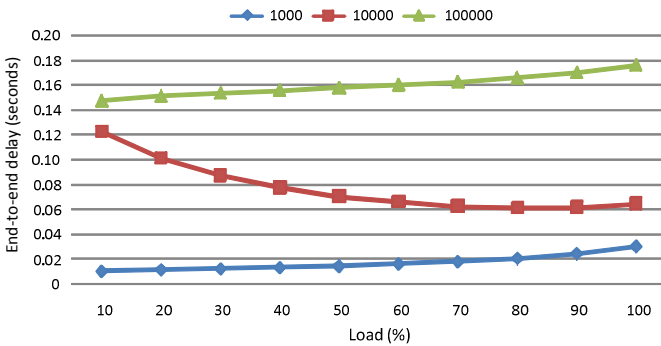


Fig. 12. End-to-End delay when varying Burst size (KB)

Fig. 12 shows end-to-end delay when varying maximum burst size in the OBS part of OBWMA. We set the maximum burst time to a large value (e.g., 10 s) to measure the real impact of burst size on end-to-end delay. We consider values: 1000, 10000 and 100000 KB. As expected, the more the burst size is larger, the more the delay is bigger. However, differently from the other values, when burst size is 10000 KB, delay tends to decrease when the load increases. We can say that at this value of burst size, delay is sensitive to the traffic intensity since the ratio [WMN packet size / OBS burst size] is moderate in this case. Indeed, with burst size 10000 KB, the ratio is roughly 1/10 (recall that default WMN packet size is 1000 KB in our simulations). For the other burst size values, the ratio is 1/1 and 1/100 for burst size 1000 KB and 100000 KB, respectively.

7 Conclusion

In this paper, we proposed a novel MAN architecture, called Optical Burst Wireless Mesh Architecture (OBWMA). OBWMA uses a set of wireless mesh networks for the access and an optical burst switching network as a backbone core network. To guarantee QoS in OBWMA, we developed an analytical model to compute end-to-end

delay and we proposed a novel adaptive burst assembly scheme (AHBA) for OBWMA. Also, we proposed a Control Bridge (CB) that coordinates QoS mapping at the border between WMN and OBS parts of OBWMA. Simulation results using ns-2 simulator showed the feasibility of OBWMA architecture, the accuracy of the proposed delay analytical model and the relevance of the proposed burst assembly scheme (AHBA).

In future work, we plan to consider bandwidth provisioning in the WMN and the OBS parts of OBWMA.

References

1. Chi, Y., Zhengbin, L., Anshi, X.: Dual-Fiber-Link OBS for Metropolitan Area Networks: Modelling, Analysis and Performance Evaluation. In: Proceedings of IEEE Globecom 2008 (2008)
2. Akyildiz, F., Wang, X., Wang, W.: Wireless Mesh Networks: A Survey. *Journal of Computer Networks* 47(4), 445–487 (2005)
3. Qiao, C., Yoo, M.: Optical burst switching (OBS) - a new paradigm for an optical Internet. *Journal of High Speed Networks* 8(1), 69–84 (1999)
4. Trigila, S., Lucidi, F., Raatikainen, K.: A service architecture for fixed and mobile convergence. *Computer Communications* 25(2), 133–148 (2002)
5. Luo, Y., Wang, T., Weinstein, S., Cvijetic, M., Nakamura, S.: Integrating Optical and Wireless Services in the Access Network. In: Proceedings of OFC 2006 (2006)
6. Luo, Y., Ansari, N., Wang, T., Cvijetic, M., Nakamura, S.: A QoS Architecture of integrating GEPON and WiMAX in the access network. In: Proceedings of IEEE Samoff Symposium 2006 (2006)
7. Wang, T., Junqiang, H., Suemura, Y., Nakamura, S.: Optical Wireless Integration at Network Edge. In: Proceedings of COIN-NGNCON 2006 (2006)
8. AlSabbagh, H.M., Chen, J., Qian, W.: A Simple Paradigm for Supporting the New Generation of Internet Based on WLAN over OBS. In: Proceeding of ICWMC 2007 (2007)
9. Martinez-Yelmo, I., Soto, I., Larrabeiti, D., Guerrero, C.: A Simulation-Based Study of TCP Performance over an Optical Burst Switched Backbone with 802.11 Access. In: Pras, A., van Sinderen, M. (eds.) EUNICE 2007. LNCS, vol. 4606, pp. 120–127. Springer, Heidelberg (2007)
10. Rezgui, J., Hafid, A., Gendreau, M.: A distributed admission control scheme for Wireless Mesh Networks. In: Proceedings of BROADNETS 2008 (2008)
11. Perkins, C., Bhagwat, P.: Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers. In: Proceedings of ACM SIGCOMM 1994 (1994)
12. Cao, X., Li, J., Chen, Y., Qiao, C.: Assembling TCP/IP Packets in Optical Burst Switched Networks. In: Proceedings of IEEE Globecom 2002 (2002)
13. Belbekkouche, A., Hafid, M., Gendreau, M.: A Reinforcement Learning-Based Deflection Routing Scheme for Buffer-Less OBS Networks. In: Proceedings of IEEE Globecom 2008 (2008)
14. Vokkarane, M.K., Haridoss, K., Jue, J.P.: Threshold-based burst assembly policies for QoS support in optical burst-switched networks. In: Proceedings of SPIE OptiComm 2002 (2002)
15. NS-2 simulator, <http://www.isi.edu/nsnam/ns>
16. Yijun, X., Vandenhouste, M., Cankaya, H.C.: Control architecture in optical burst-switched WDM networks. *Journal on Selected Areas in Communications* 18(10), 1838–1851 (2000)

Evaluation of a QoS-Aware Protocol with Adaptive Feedback Scheme for Mobile Ad Hoc Networks

(Short Paper)

Wilder Castellanos, Patricia Acelas, Pau Arce, and Juan C. Guerri

Multimedia Communication Group, ITEAM Institute, Technical University of Valencia,
Camino de Vera, s/n, 46022, Valencia, Spain
wilcashe@posgrado.upv.es, {patacdel, paarvi}@iteam.upv.es,
jcguerri@dcom.upv.es

Abstract. Due to bandwidth constraint and highly dynamic topology in ad hoc network systems, one of the major challenges is the deployment of end-to-end quality-of-service support mechanisms. Time-sensitive communications like video applications may be seriously disrupted if these QoS support mechanisms don't exist. In this paper we propose a QoS routing protocol based on AODV (AQA-AODV), which creates routes according to application QoS requirements. We have introduced link and path available bandwidth estimation mechanisms and an adaptive scheme that can provide feedback to the source node about the current network state, to allow the application to appropriately adjust the transmission rate. The simulation results reveal the performance improvements in terms of packet loss and delay while the end-to-end throughput is not affected compared with the throughput achieved by other protocols like AODV.

Keywords: Wireless ad hoc networks, quality-of-service aware routing, AODV, Bandwidth estimation.

1 Introduction

A mobile ad hoc network is a group of autonomous wireless devices organized themselves dynamically in a mesh topology. The key feature of this type of networking is the nonexistence of any permanent infrastructure. Perspective video communication over such networks can be expected in various scenarios, both in civil and military activities. However, hard communication conditions because the wireless channel is shared among adjacent hosts and network topology can change as hosts move, do intensify challenges against transmission of video packets. Especially when video applications generate a huge data volume that is delay-sensitive, bursty and loss of some important data segments, such as synchronization data, may seriously disrupt a long sequence of frames [1]. Additionally, to maintain an acceptable playback quality, excessive communication delay is not tolerated.

The main issue is how to efficiently transmit a large volume of time-sensitive data given that many packets are dropped due to the fact that network resources are limited and time-varying.

We propose a strategy based on a QoS-aware routing protocol that allows the source to adapt the transmission rate. Our protocol AQA-AODV (Adaptive QoS-Aware AODV) is a modified and enhanced version of the Ad hoc On-demand Distance Vector [2] (AODV). More precisely, we have introduced link and path available bandwidth estimation mechanisms and an adaptive feedback scheme into the original AODV protocol. Similar mechanisms are studied in [3][4]. In addition, a QoS extension is added to the AODV control packets and the routing table. The only QoS metric considered in our solution is bandwidth for a QoS flow, because finding a route subject to multiple metrics in many cases is considered to be an NP-complete problem [4]. The result is a QoS path finding mechanism that can provide feedback to the application about the current network state to allow the application to appropriately adjust the transmission rate.

In order to test the performance of our protocol we have implemented the proposed protocol in the NS-2 simulator. Results indicate that the packet loss and packet delay decrease significantly, while the overall end-to-end throughput is not impacted, compared with routing protocols that do not provide QoS support.

This paper is organized as follows. Section 2 describes the impact of packet forwarding over delay, packet loss and channel capacity in wireless ad hoc networks for very simple linear topology with very regular traffic patterns. Section 3 describes our proposed QoS-aware routing protocol that incorporates QoS into Ad hoc On-demand Distance Vector (AODV). Section 4 presents the performance evaluation of our QoS-aware routing protocol and Section 5 offers some conclusions.

2 Capacity of Ad Hoc Wireless Networks

One of the limitations of wireless ad hoc networks is the achievable capacity due to the fact that nodes cannot simultaneously access the shared medium. More specifically, when a node is transmitting a packet, neighbor nodes within its Interference Range (IR), have to keep silent. This fact degrades the wireless data rate.

This section examines the feasible capacity of a well known single linear network topology of nodes where the source of traffic is the first node, destination is the last node and the packets are being forwarded through the intermediate nodes. The source node sends data as fast as its MAC allows it. In this scenario only adjacent nodes are in transmission range of each other. A more detailed study is presented in [6]. The results of our simulations are shown in the Fig 1. The simulation suggests that capacity along chain can be surprisingly low. We can see that, when the source node is sharing the channel with only 1 node, the throughput could reach up to 1.4 Mbps for 1000 bytes/packets, due to the overhead produced by RTS, CTS and ACK packets. When the hop count is increased, the maximum throughput of one flow is decreased substantially and falls down until 0.2 Mbps due to the overhead of MAC layer and the mutual interference between packets of the same flow, also called "Intraflow contention" [7].

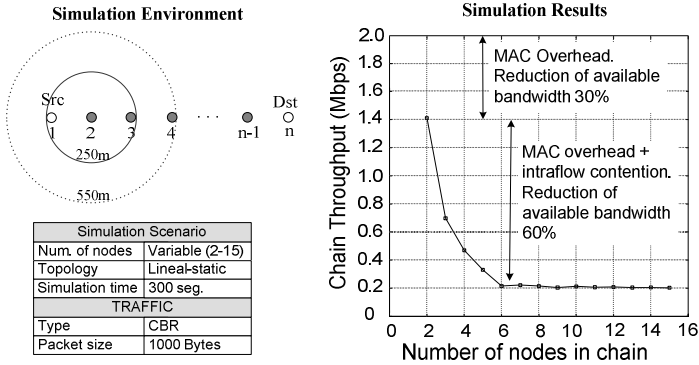


Fig. 1. Relationship between maximum throughput and number of nodes along a linear network topology

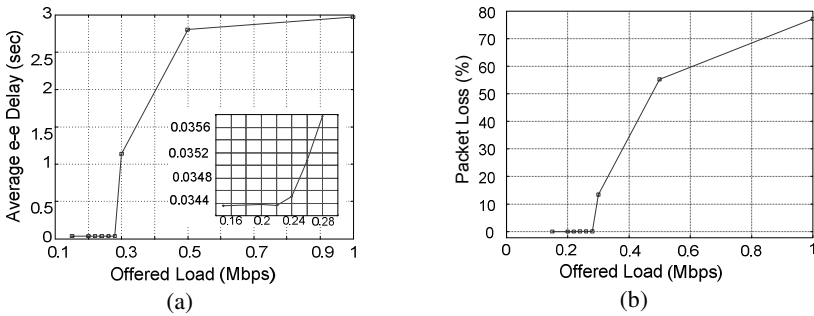


Fig. 2. Average end-to-end Delay (a) and Packet Loss (b) over linear topology of 7 nodes

For a linear topology of 7 nodes where the source of traffic is the first node and the destination is the node 7, the packet losses are increased significantly when the maximum throughput that can reach a chain of 7 nodes (about 0.20 Mbps) is slightly exceeded by the source. Fig. 2 shows the increase of the end-to-end delay and packet losses, as the source increases its transmission rate.

These results show that the relation between ad hoc routing and capacity suggests that any evaluation or implementation of a wireless ad hoc network requires an understanding of network capacity. In particular, 802.11 MAC interacts with forwarding impact over the end-to-end delay and packet loss, both important metrics for video transmission over wireless ad hoc networks.

3 QoS-Aware AODV Protocol with Adaptive Feedback Scheme

Our proposed routing protocol called AQA-AODV (Adaptive QoS-Aware Ad-hoc On-demand Distance Vector), is an AODV-based protocol. Our key modifications include new fields in RREQ and RREP packets to the bandwidth requirements and a “session ID”, used to identify each QoS flow that is established. The solution in reference [8] is

an admission control based protocol. However, available bandwidth estimation and consumed bandwidth prediction algorithms are not defined. An important difference between our proposed protocol and other AODV-based solutions is the adaptive feedback scheme by which the source node can easily adapt its transmission rate according to the state of the route. For this reason, nodes along the path must know their available resources (in terms of bandwidth) by using some algorithms.

3.1 Route Discovery in AQA-AODV

For route discovery, if a source node requires a route to a destination node with specific bandwidth requirements, it broadcasts a RREQ packet with the QoS extension to its neighbor nodes. The RREQ packet is rebroadcasted as in AODV until the RREQ packet reaches the destination node [2]. Only the destination will be able to send the route reply. This will ensure that all nodes in the selected route satisfy the bandwidth constraints. When the destination node receives a RREQ packet, before sending the RREP to the source, local available bandwidth is checked and estimation of the intraflow contention is necessary, by using the relation between the number of hops and the end-to-end throughput. This will allow the node to estimate the bandwidth along a path while taking into account the contention between packets of the same flow. Finally, the RREP will be transmitted to the source with a modified header that includes the minimum value between required bandwidth for the source and the maximum bandwidth that all hosts along the route could support. Once an intermediate node receives the RREP packet, it compares its available bandwidth with the bandwidth indicated in the RREP. If its local available bandwidth is lower, it updates the min-bandwidth field in RREP, using its available bandwidth. Otherwise, the node forwards the RREP. This procedure will ensure that the source knows the min-bandwidth along the path which will be the maximum rate that it may transmit. Fig. 3 illustrates the overall operation of the key phases of AQA-AODV.

3.2 Estimation of the Available Bandwidth in AQA-AODV

AQA-AODV uses a similar method to one presented in [9] to determine the available bandwidth at a node. To estimate the available bandwidth, BW_{av} , nodes simply add up the size of packets sent, received and detected in a fixed period of time T . The channel bandwidth when transmitting a packet is calculated using the equation (1).

$$BW_{av} = \frac{S}{T_r - T_s} \quad (1)$$

In (1) $S = RTS+CTS+HELLO+ACK+RTS+CTS+HELLOACK$, i.e. the size of all packets (in terms of bits) sent from the source to destination during the period T , where T is equal to $T_r - T_s$. T_s is the time when the data packet is ready to be sent at the source, while T_r is the time when the ACK for the data is received at the source.

With the information of the available bandwidth at the nodes, it is still not simply to compare the available data rate at node and the required data rate for one traffic when deciding whether the node satisfy the requirement, it has to check if the given flow fits or not into the n -hop route. The method to provide an estimation of the consumed bandwidth along the route used in AQA-AODV was adapted from [10].

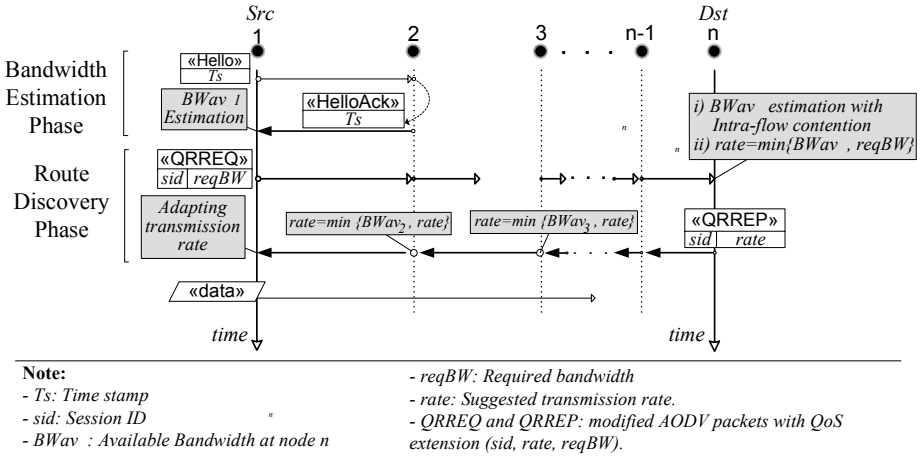


Fig. 3. Overview of AQA-AODV

4 Performance Evaluation

Network Simulator (NS-2) has been used to test the performance of our QoS-aware routing protocol. This simulator implements the IEEE802.11 protocol for the MAC layer, working in the Distributed Coordination Function (DCF) mode with a channel data rate of 2 Mbps. The transmission range and interference range are 250 m and 550 m respectively.

The performance of our QoS-aware routing protocol was evaluated by comparing it with conventional AODV, which has no QoS support, using two simulation scenarios. The first scenario consists of a chain of nodes where the performance was evaluated as a function of the chain length. The second scenario consists of 30 nodes move in a 1000m x 1000m area according to the random waypoint model with pause time set to 20sec. The nodes move toward a random destination using a speed randomly chosen between 0 – 3 m/s. A random source-destination pair send packets using a request rate between 0,1 and 1,0 Mbps. All traffic flows are Constant Bit Rate (CBR) streams over UDP with a packet size of 1000 bytes.

4.1 Simulations Results

We evaluated the performance of AQA-AODV by measuring three parameters: end-to-end data packet delay, packet loss and the maximum throughput achieved along the route. Each data point shown in the figures is the average of 10 simulations with different random seed. The results are presented as follows, according to the aforementioned scenarios.

In the first scenario (static linear topology with variable length), the performance of AQA-AODV is tested as function of the number of hops on the path. Node 1 is the source of data traffic and the last node in the chain is the traffic sink. Fig. 4a shows the ability of our Adaptive CBR source to adjust its rate according to network status and according to the number of competing nodes. Initially the source required a

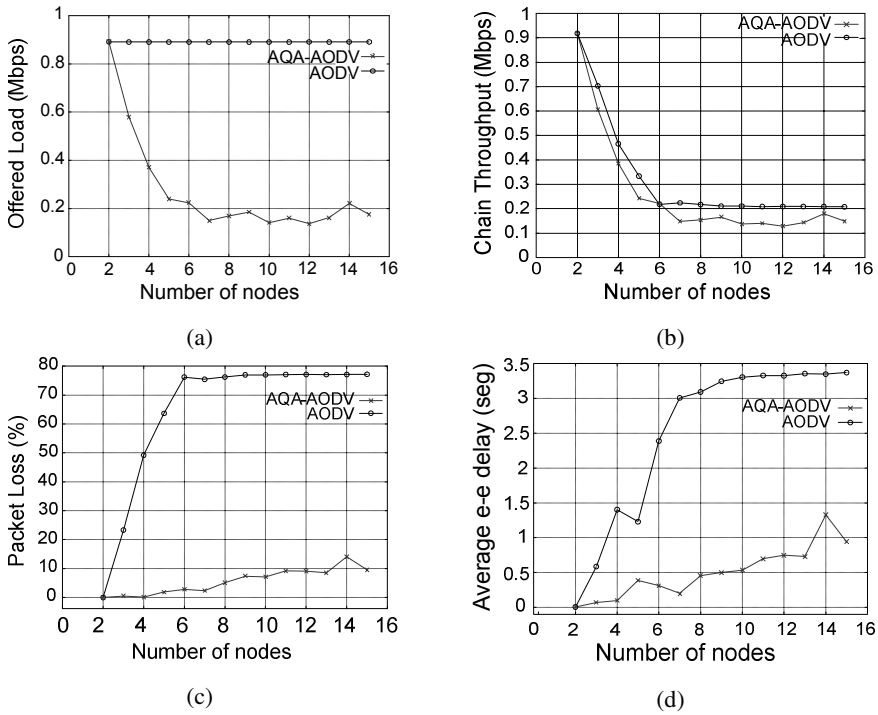


Fig. 4. Transmission rate (a), Throughput achieved along the path (b), Packet Loss (c), Average end-to-end delay (d) with variable chain length

transmission rate of 0,9 Mbps. In a 2-nodes chain it is possible without leading network congestion. However, when chain has 3 or more nodes, this transmission rate is not supported efficiently. Therefore our adaptive source adjusts its data rate. While using AODV, the source sends packets to a fixed rate of 0,9 Mbps.

As seen in Fig. 4b the total network throughput achieved with AQA-AODV is very close to throughput achieved using AODV. However, using AQA-AODV, the network congestion is significantly reduced. Therefore, the time used for waiting in the packet queue and contending for the channel decreases. In other words, our adaptive feedback scheme allows getting an important decrease in packet loss (Fig. 4c) and delay (Fig. 4d) without any bandwidth sacrifice.

Fig 5 shows the results of our simulations in the mobile topology. In terms of packet loss, AQA-AODV shows great improvement over AODV, which achieves very high packet losses for some requested rates. For example, the packet loss is between 19% and 83% using AODV, whereas using AQA-AODV the packet loss remains lower than 24%. Fig. 5b shows that the average end to end delay of AQA-AODV is always below 0,4s, whereas, the end to end delay of AODV increases badly when the transmission rate increases from 200 kbps to 1000 kbps. With AODV, the maximum average end to end delay reaches 1,9s at 700 kbps, about 16 times higher than using AQA-AODV. As seen in Fig. 5c the total network throughput achieved with AQA-AODV is very close to throughput achieved using AODV. We would

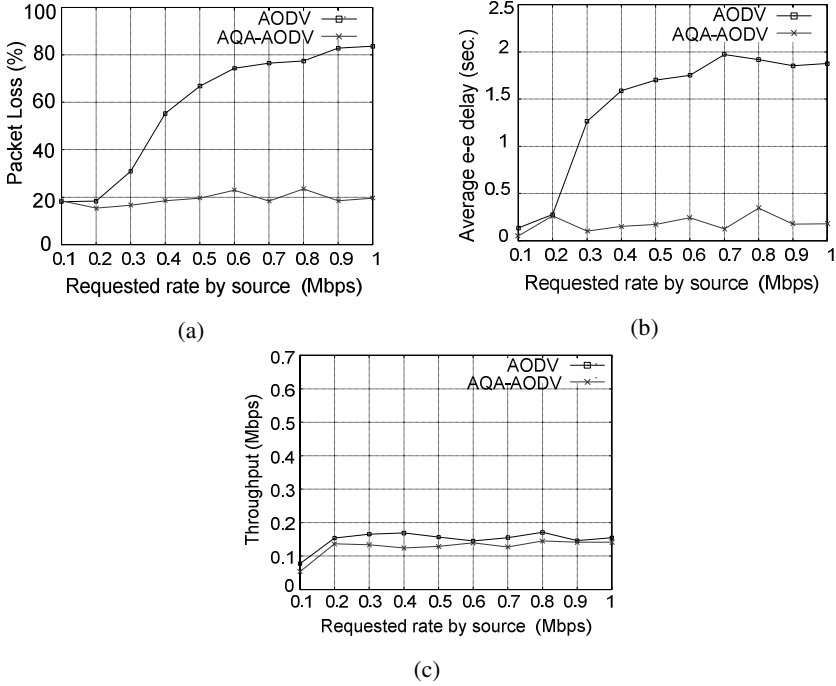


Fig. 5. Packet Loss (c), Average end-to-end delay (b) and Throughput (c) with variable requested rate

expect that the AQA-AODV protocol’s performance will degrade in scenarios with high mobility because the nodes will need a specific time for exchange information about the network status.

5 Conclusions

A novel QoS-aware routing protocol (AQA-AODV) is proposed in this paper for carrying out time-sensitive communications over wireless ad hoc networks. AQA-AODV can avoid network congestion by a simple and precise cross layer approach with adaptive feedback scheme to provide information to the application about the network status. Our protocol incorporates bandwidth estimation through Hello packets and a prediction of consumed bandwidth that take into consideration the interference between packets of the same flow. Simulations show that this proposed scheme could reduce significantly both the dropping rate and the end-to-end delay without impact the overall end-to-end throughput.

In the future, we plan to examine how to implement a predictive way to foresee a route break, which would avoid performance degradation in mobile environment. Hence, methods such as pre-emptive maintenance routing [11] might be implemented to help the routing protocol to reduce the transient time when the required QoS is not guaranteed due to a route break. Also, our future work includes a mechanism to

periodically check that the available bandwidth is still available. In this paper, in terms of metrics used in the QoS extension, only data bandwidth is considered in the simulations. End-to-end delay could be added during the route discovery and maintenance in the routing protocol.

Our ultimate goal is to provide a cross-layer framework where the video source exploits the feedback information from the underlying protocol (AQA-AODV) to tune a parameter on the source coding to adapt the traffic rate to the path.

References

1. Wu, D., Hou, T., Zhu, W., Lee, H.-J., Chiang, T., Zhang, Y.-K., Chao, H.J.: On End-to-End Architecture for Transporting MPEG-4 Video Over the Internet. *IEEE Transactions on Circuits and Systems for Video Technology* 10(6), 923–941 (2000)
2. Perkins, C., Royer, E.M., Das, S.: Ad hoc on-demand distance vector (AODV) routing, IETF RFC 3561 (2003)
3. Gerasimov, I., Simon, R.: A bandwidth-reservation mechanism for ondemand ad hoc path finding. In: *Proc. IEEE Simulation Symp.*, pp. 27–34 (2002)
4. Chen, L., Heinzelman, W.B.: QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications* 23(3), 561–572 (2005)
5. Wang, Z., Crowcroft, J.: Quality-of-service routing for supporting multimedia applications. *IEEE J. Select. Areas Commun.* 14(7), 1228–1234 (1996)
6. Li, J., Blake, C., Couto, D.D., Lee, H., Morris, R.: Capacity of ad hoc wireless networks. In: *Proc. 7th ACM Int. Conf. Mobile Comput. Netw.*, pp. 61–69 (2001)
7. Sanzgiri, K., Chakeres, I., Belding-Royer, E.: Determining intra-flow contention along multihop paths in wireless networks. In: *Proc. Broadnets 2004 Wireless Netw. Symp.*, pp. 611–620 (2004)
8. Perkins, C., Royer, E.M.: Quality of service for ad hoc on-demand distance vector routing. IETF Draft (2004), <http://www.psg.com/~charliep/txt/aodvid/qos.txt>
9. Xue, Q., Ganz, A.: Ad hoc qos on-demand routing (aqor) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing* 63(2), 154–165 (2003)
10. De Renesse, R., Friderikos, V., Aghvami, H.: Towards Providing Adaptive Quality of Service in Mobile Ad-Hoc Networks. In: *IEEE VTC 2006, Melbourne* (2006)
11. Goff, T., Abu-Ghazaleh, N.B., Phatak, D.S., Kahvecioglu, R.: Preemptive maintenance routing in ad hoc networks. In: *Proc. MobiCom*, pp. 43–52 (2001)

QShine 2009

**Session III – Query and Coverage Issues
in Sensor Networks**

Adaptive Data Quality for Persistent Queries in Sensor Networks

Vasanth Rajamani and Christine Julien

Department of Electrical and Computer Engineering
The University of Texas, Austin
{vasanthrajamani, c.julien}@mail.utexas.edu

Abstract. Wireless sensor networks are emerging as a convenient mechanism to constantly monitor the physical world. The volume of information in such networks can be extremely large. And, to be meaningful to applications, this information must be processed at the right level of accuracy. However, there is an inherent trade-off between achieving a high degree of data accuracy and the communication overhead associated with achieving it. We present a simple mechanism for spatially approximate query processing. We present a protocol that leverages gossip based routing to collect network data from a randomly selected set of nodes at a user-defined level of accuracy. We extend this protocol to address persistent queries, long running queries where network data is collected periodically, by treating a persistent query as a temporal aggregate of individual queries. Finally, we provide a novel protocol that dynamically adapts its accuracy based on the quality of the responses to individual requests in the persistent query. We describe this protocol in detail and evaluate its performance through simulation.

Keywords: Adaptive fidelity, gossip.

1 Introduction

Sensor networks have been deployed in a wide range of applications that monitor the physical world in real time such as intelligent construction sites, habitat monitoring, and industrial sensing [19]. When sensor networks are deployed on a large scale, however, there is an explosion in the amount of data to observe and analyze. Requirements on the quality of data collected varies by application and environmental changes. For example, a foreman might evaluate the safety of a construction site by observing the movement of equipment in the site. A periodic summary of data in the site may suffice until a construction truck moves into the danger circle of a crane. At this point, more information should be obtained and the worker should be warned about the danger in his environment. The data fidelity should be increased only when behavior of interest is detected. In this paper, we present a technique that automatically tunes the fidelity of data based on user specifications for such applications.

We address two types of queries that are of value in sensor networks: *one-shot queries* and *persistent queries*. A one-shot query is a one-time occurrence

in which the application requests data values from some or all of the nodes in the network. This query has no relationship to other queries that may be issued by the application. On the other hand, a persistent query is a long-lived operation that provides periodic responses. We implement persistent querying as a temporal aggregation of component one-shot queries. Doing so allows us to use results of component one-shot queries to influence the behavior of subsequent component queries.

Sensors are typically battery operated and hence extremely resource-constrained. Communication costs account for a large amount of the battery drain during operation, and sending large amounts of unnecessary data reduces the network lifetime substantially. Two approaches have been proposed to reduce the query communication overhead—in-network aggregation (e.g., [14]) and approximate querying (e.g., [17]). In-network aggregation techniques typically build and maintain a tree over the network and distribute the aggregation operation along all non-leaf nodes of the tree. In approximate query processing, the response is typically provided to the user as an estimation of the correct answer with deterministic or probabilistic guarantees quantifying the confidence in the estimate. Both, in-network aggregation and approximate query processing (AQP) have some attractive properties. In-network processing uses actual data collected from sensors. AQP typically models the data at a base station and periodically updates the stored model using values from the network [4]. Other approaches [20] form spatially correlated groups, and one node per group participates in the query resulting in fewer messages being transmitted.

In networks with dynamic data values, it is beneficial to retrieve actual data values and be less reliant on an *a priori* model. In this paper, we present a querying technique that provides approximate querying by selecting a subset of nodes and using actual data from these nodes at query time. This frees our approximation protocol from maintaining state information on the nodes and also alleviates the need for all nodes to participate in every query. While it is possible to accomplish this approximation using any number of underlying networking protocols, we choose gossip routing [3,11]. In its basic form, on receiving a packet, a node chooses to retransmit or drop a packet based on a threshold parameter, p . When a node receives a query, if it chooses to retransmit the query, it actively takes part in resolving the query; otherwise it drops the packet and reduces the likelihood that its downstream neighbors participate. Such a protocol is inherently approximate, as the number of nodes participating varies probabilistically depending on parameter p . In addition to adapting itself nicely to AQP at a conceptual level, gossip routing is robust to changes. Current in-network aggregation and approximate algorithms tend to maintain a tree or a cluster based overlay in which the failure of an intermediate node can lead to significant overhead in rebuilding the aggregation framework. Gossip routing does not impose any hierarchy on the network, and the overhead of performing successive queries is impacted less by the node failures that are fairly common.

Another advantage of using gossip routing is that it allows us to make no assumptions about the data or network characteristics. Some approximation

algorithms cluster nodes with highly correlated sensor values [20] which requires *a priori* knowledge of the range of data values in the network. Since we make no assumptions about the data or network characteristics, we instead associate *data quality metrics* with query responses. These metrics may include the number of nodes participating, the variance of sampled data, and the spatial distribution of the sampled nodes. For one-shot queries, the user can use this quality metric as a yard stick by which to analyze his results [15]. For persistent queries, where the query measures sensed conditions over a period of time, we use these data quality metrics to adapt the query's intended fidelity automatically. In this paper, we present a mechanism to perform *adaptive approximate query processing*. Because we view a persistent query as an aggregate of one-shot queries, we use the data quality metrics associated with individual one-shot queries to adapt the fidelity of the protocol for subsequent one-shot queries. Sensor network queries can be broadly classified into two types— aggregate and stream queries. Aggregate queries provide a single aggregated answer, like the average value of the sensor network. Stream queries typically return a stream of data values from different nodes in the network. In this paper, we focus on providing a protocol for adaptive approximate query processing for aggregate queries. Previous work on gossiping [11, 5, 16] has focused on computing aggregates in a completely distributed fashion. In contrast, we employ gossiping to retrieve a subset of data and control the degree of hosts involved by analyzing the collected data (answers to component queries in the persistent query) in a centralized manner.

The novel contributions of this work are as follows. First, we propose a protocol that incorporates gossip routing for spatially approximate query processing. Second, we discuss the impact of exposing various data quality metrics and underscore how an application can use them to interpret the quality of a response. Third, we show how to incorporate data quality metrics to adapt the accuracy of the AQP algorithm for persistent queries. Finally, we evaluate our protocols and verify their utility.

The rest of this paper is organized as follows. Section 2 adapts gossip routing to AQP. Section 3 evaluates gossip routing for the task of AQP. Section 4 provides a mechanism to expose data quality and uses it to provide context. Using the insights gained from evaluating the AQP protocol, Section 5 provides a protocol that performs adaptive AQP, and Section 6 evaluates its performance. Section 7 discusses related work, and Section 8 concludes.

2 Gossip Routing Based AQP

In this section, we describe how gossip routing provides approximate responses to applications' queries. Gossip routing is based on probabilistic broadcasting, in which a predetermined threshold, p , determines whether a node rebroadcasts or drops a received packet. If p is one, then the behavior is equivalent to flooding. In most networks, setting p to a value smaller than one can still result in a packet reaching all nodes in the network with a very high probability. In gossip routing, only a subset of nodes are involved in query execution. If the query is executed

several times, this subset is likely to be different each time, thereby spreading the load more evenly. However, since all nodes in the network do not participate in every query, the result obtained is inherently approximate. In the rest of this section, we explain how we adapt gossip routing to suit our needs.

We first evaluate a basic gossip protocol for providing approximate query results. The state variables for each host in our protocol are shown in Fig. 1. Only the state for a single query is shown; each query has a duplicate set. Our protocol uses two types of packets: *Query* packets and *QueryReply* packets. A *Query* packet looks like:

<i>id</i>	– A’s unique host identifier
<i>neighbors</i>	– A’s logically connected neighbors
<i>parent</i>	– A’s parent in the tree
<i>p</i>	– A’s probability threshold for broadcasting an incoming query
<i>data</i>	– A’s data value obtained from its sensors

Fig. 1. State Variables for Protocol on Node A

$$\langle query_id, p, data_request, sender, originator \rangle.$$

The *query_id* is used to ensure a node does not respond to or forward the same query twice. The *p* value is the probability with which a receiving node should retransmit the packet. The *data_request* contains the application’s data needs (e.g., the type of sensor reading desired). The *sender* of a query is the node that forwarded the packet, while a query’s *originator* is the query issuer. A *QueryReply* packet simply contains the data that is the response, the unique query id number, and the id of the destination host (i.e., the query issuer):

$$\langle query_id, data, destination \rangle.$$

These two packet types are kept necessarily simple to accommodate resource-constrained networks. To define the protocol’s behavior, we use I/O Automata notation [13]. We show the behaviors of a single host, A, indicated by the subscript A on each behavior. Each *action* (e.g., *QueryReceived_A(q)* in Fig. 2) has an effect guarded by a precondition. Actions without preconditions are *input actions* triggered by another host. Each action executes in a single atomic step. We abuse I/O Automata notation slightly by using, for example “send *Query* to *neighbors*” to indicate a sequence of actions that triggers the QUERYRECEIVED action on each neighbor.

The basic gossip protocol is very simple. When a node receives a query, it first logs the query’s sender (*q.sender*) as its parent and sends its sensor data through its parent to the query issuer. It then uses the probability *p* to determine whether it will forward this query to its neighbors. To prevent nodes from processing the same query multiple times, a node checks whether it has received the query previously (based on the query’s unique id) before processing it. Fig. 2 shows this behavior in I/O Automata form. Also shown in the figure is a node’s behavior in response to receiving a *QueryReply*; the node checks if

it is the targeted destination; if not, the node forwards the packet to its parent. This is a slightly simplified version of the protocol that only considers a single query from a single application that is active at a given time in the network. The protocol's implementation maintains additional state to sort out the forwarding information associated with different active queries.

```

QUERYRECEIVEDA(q)
  Effect:
    if !received(q.query_id) then
      parent := q.sender
      r = (q.query_id, data, q.originator)
      send QueryReply(r)
      p := rand()
      if p ≥ q.p then
        q.sender = A
        send Query(q) to neighbors
      end
    end
  end

QUERYREPLYRECEIVEDA(r)
  Effect:
    if r.destination = A then
      /** send r.data to application ***/
    else
      send QueryReply(r) to parent
    end
  end

```

Fig. 2. Gossip based AQP

this example, nodes *f* and *h* decide to drop the query. They each create a reply packet containing their node identifier and data value and send it back to the client. (The query identifier has been omitted from the figure for brevity.) Based on the probability *p*, nodes *c* and *g* choose to forward the query. Both create reply packets they send back to the client. Node *c* forwards the query to *b* and *d*. When node *c* receives a reply from *b* or *d*, it simply forwards it to its parent, the querier.

An application can aggregate the collected values or use individual readings depending on its needs. A biologist checking the number of animals in the habitat might be satisfied with an aggregate sum value. On the other hand, he might want to build a map of their movement by using information from all the data in his query response. Each query might probabilistically choose a different set of sensors and, over a period of time, the biologist can build a complete picture without having to query all the nodes every time. Since imprecision is an inherent part of any approximate algorithm, we expose data quality metrics to provide context to the query response. A very simple data quality metric is the number of nodes that participated in the query. Consider a query where the user is

Fig. 3 illustrates the protocol with an example. The value inside a circle is a node's data value. The client is the node in the center—the circle containing a PDA. In the figure, the dashed lines correspond to query replies, while the thick lines indicate a query being forwarded. The tuples next to the response line show the sensor readings carried by the query reply packets. Nodes with concentric circles dropped the packet. On receiving a query from the application, the client broadcasts it to its neighbors {*c*, *f*, *g* and *h*}. The query contains a probability threshold specified by the application. In

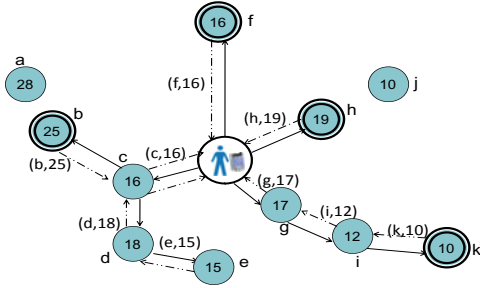


Fig. 3. Protocol example

interested in sensor readings from nodes placed on cranes in an industrial site. If he receives a stream response with just two crane values, when there are 10 cranes visible, he might choose to reissue the query with a higher p . While counting the number of animals, the biologist will feel a lot more secure about the query response he receives if he is provided a confidence interval along with the average humidity. This can be done by exposing the data’s variance and the number of nodes sampled. In the rest of this paper, we focus on situations where it is beneficial to aggregate the data values returned from the sensor nodes. This protocol works on the assumption that the client device provides the probability p as an input. In the next few sections we show that changing p does in fact affect the accuracy of the results. We also show how this simple protocol can be leveraged to adaptively tune the accuracy of the result for persistent queries by using the data quality metrics exposed.

3 Effectiveness of Using Gossip Routing for AQP

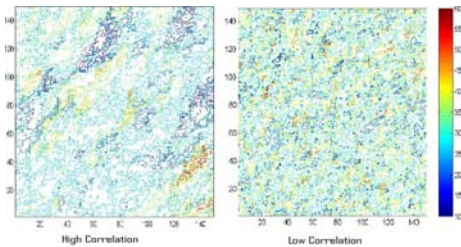


Fig. 4. Correlation of Sensor Grids

Our gossip protocol assumes that, given a good value of p , one can leverage gossip routing to perform approximate query processing. In this section we study the impact of changing p and ascertain whether this assumption holds in different environments. We also develop insights on how to manipulate p to accommodate different application requirements. To thoroughly evaluate

our protocol, we used the TOSSIM network simulator [12], which allows direct simulation of TinyOS [6] code written for MICA2 motes.

3.1 Data Set

For modeling sensor data we used a tool provided by Jindal and Psounis [9]. The tool generates spatially correlated synthetic data for sensor networks of varying sizes. The data traces generated have been shown to be very close to physical phenomenon observed in the real world. Since our goal is to investigate the feasibility of using gossip routing to perform approximate query processing, this tool provides us a convenient way to test under different simulated environments. We generated spatially correlated sensor data for a 150m x 150m grid. The tool takes in an input parameter β which allows us to manipulate the degree of spatial correlation in the sensor network. A higher value of β makes a node more likely to choose a data value independent of its neighbors', thus producing spatially uncorrelated data. We varied the value of β to 0.001, 0.018 and 0.33, producing sensor networks with high, medium and low data correlations respectively. Fig. 4 shows example surface plots of two data traces we used; the plot on the left shows highly correlated data, while the right shows a data trace that is significantly less correlated.

3.2 Simulation Setup

We generated synthetic traces corresponding to the distribution of temperature data in a sensor network field. Given the synthetic data, we explore whether using gossip routing is effective in performing approximate query processing. The protocol is used to query data from the network and compute the value of the average temperature. We used a uniform random placement of sensor nodes. To model the radio transmissions, we used TOSSIM's disc model with a radius of 10 feet. The number of sensor nodes in the network was set to 100. Error bars indicating 95% confidence intervals are included in the graphs whenever possible.

We posit that increasing p will increase the accuracy of our protocol. Fig. 5 confirms that the relative mean error across all responses does decrease as p becomes larger. The accuracy increases because the number of nodes responding with a data value increases as p gets larger. It increases from about 5 to 23 nodes as p varies between 0.1 and 1. The figure plots the normalized error against different values of p for data sets with three different correlation levels. The absolute error is the difference between our protocol's computed average and the actual average provided by an oracle. The normalized error is the absolute error normalized by the correct average provided by an oracle. The normalized error decreases as p increases, regardless of the data distribution, although the

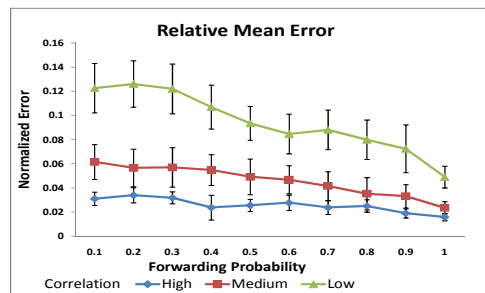


Fig. 5. Accuracy as a function of p

¹ The source code for the tool is provided at <http://www-scf.usc.edu/~apoorvaj/tool.html>

decrease is much more pronounced when the underlying data is less correlated. This is intuitive because, when all nodes have more or less the same data value, sampling more nodes will not produce a large change in the final answer. We can infer from this graph that using a large p is of limited value when the data is highly correlated. Even when querying very few nodes, by setting p to a low value, a gossip protocol can produce an answer very close to the correct response. This is one of the key insights we use in Section 5 to automatically adapt the protocol for persistent queries even when there is no.

However, increasing the number of nodes involved in a query comes with the overhead of increased number of transmitted messages. Fig. 6 plots the increase in overhead associated with raising the value of p . For example, when p is increased from 0.5 to 1, the number of messages transmitted increases almost seven fold. As radio transmission accounts for a large amount of the energy consumption in battery powered devices, choosing the right value of p provides a good lever to trade energy for accuracy while performing approximate query processing.

In addition to the number of nodes participating, an important factor to consider is the spatial distribution of the nodes that respond. Fig. 7 shows the spatial distribution of the nodes participating in a query. The bins on the horizontal axis represent the distance from of the responding nodes the querier. For example 20-30 represents nodes between 20m and 30m from the query issuer. The vertical axis is the percentage of nodes that responded out of all nodes that were reachable from the querier at that distance range. When the retransmission probability is low, the response obtained is a local one, i.e., it is biased towards nodes close to the querier. However, even when the retransmission probability is high, the percentage of nodes responding to the query decreases as we move away from the querier. This effect is not only due to the value of p but is also impacted by packet collisions. This is confirmed by the fact that the percentage drops to a low value even when the network is being flooded. The farther a node is from the querier (e.g., nodes in the 80-90 bin are about 9

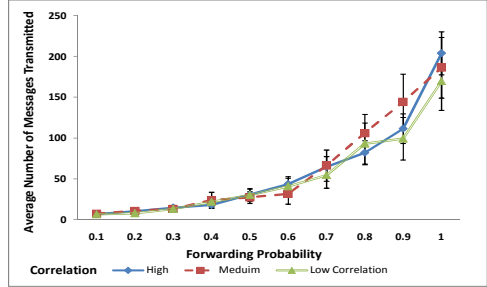


Fig. 6. Overhead vs. p

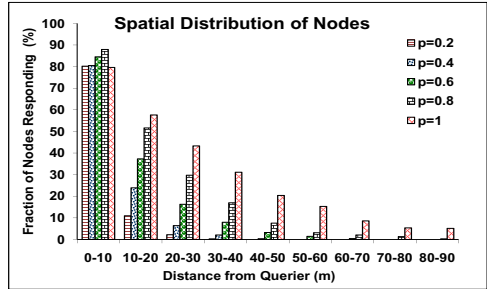


Fig. 7. Spatial Distribution of Participants

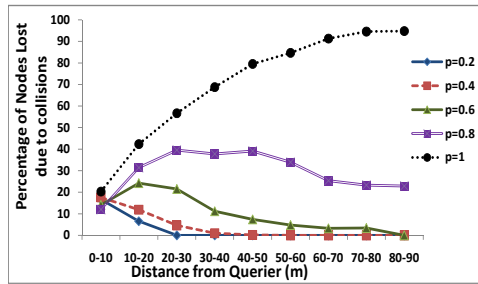


Fig. 8. Difference between Ideal and Observed Spatial Distribution

hops away on average), the greater the chance of a collision related packet drop either during query or response prorogation.

To isolate the impact of p alone on the spatial distribution of the query, we wrote a version of our protocol in Java and compared the results with the results generated using TOSSIM. Fig. 8 plots the difference between what we achieved and what we could have achieved ideally. Each data point represents 100 runs in the simulator. From the figure we can see that collisions have significant impact when the value of p is high. This behavior is acceptable for the *immersive* sensor network applications we target since the nature of the query is often localized around the query issuer. Remote distributed sensing applications, like long term industrial monitoring, require a smooth sampling of nodes by using fine-grained notions of location to route sampling messages. However, this requires the protocol to have prior knowledge of the application scenario, the physical confines of its operating area, and fine-grained location information for every node. While our approach to approximate query processing can be adapted for remote distributed sensing applications by setting a relatively high value of p , it is better suited to a vast number of *immersive* sensor network applications, that can be deployed in an ad hoc fashion to create a picture of more local data.

4 Data Quality Metrics

Imprecision is an intrinsic part of any approximate query processing system. Different quality metrics such as the number of nodes participating in the query, the variance of the underlying data, and the spatial distribution of the nodes provide the application different types and amounts of confidence when interpreting a query response. While this information can be useful for one-shot queries in helping the application determine the usefulness of the returned data [15], it can be even more beneficial to persistent queries that can adapt their querying behavior over time. We expose *data quality metrics* associated with a collective response to a query that can influence the subsequent course of action. Some example data quality metrics for aggregate sensor queries are:

- *Number of Nodes Participating (N)*: Knowing that a large number of nodes participated in the query may be sufficient to represent the quality of a returned query result. If the number of nodes is too low, an application may choose to re-issue the query with a higher p .
- *Variance (V)*: Knowing the variance across the returned data samples can also be very useful to an application. If the variance is low, the application may issue subsequent similar queries with a much lower value of p to reduce overhead. A high variance within data that is expected to be correlated indicates a poor sampling, and subsequent queries will benefit from a larger p .
- *Locality (L)*: When sensor nodes are able to attach location to their readings, exposing the data’s spatial distribution can add useful context. An easy alternative is to expose the distribution of the hop counts from the querier. This can give a good intuition for how spatially distributed are the nodes contributing to the response. A user might be able to fine tune his results successfully by just varying p . Alternatively, he may decide that the best way to get non-local results is to flood the network or use a sophisticated adaptation function (discussed in detail in Section 5), if he is interested in long term sensing style results.

These quality metrics can be exposed with very little additional computation or communication overhead. In the next section, we focus on using these metrics to dynamically tune the sequence of queries that constitutes a persistent query.

5 Adaptive Approximate Querying Protocol for Persistent Queries

We implement a persistent query as a sequence of one-shot queries. We refer to a particular one-shot query within a persistent query as a *round*. In our adaptive model, the protocol can use the data quality metrics associated with the previous rounds to parameterize the protocol’s execution for the next round. Using the data quality metrics exposed, an application developer can write an *adaptation function* that dynamically changes the behavior of a protocol for persistent queries, after considering data dynamics and user preferences. In this section we demonstrate the feasibility of our approach using an example adaptation function.

5.1 Adaptation Function

One can write complex functions in which combinations of locality and variance influence adaptation. However it becomes difficult for the application to express domain knowledge in a straightforward manner. Often, a simple function can capture the essence of the required adaptation. For example, an application that monitors chemical leaks must decide if the data obtained in any given round is significantly different from the previous rounds and change the query behavior accordingly. A good adaptation function should specify the value beyond which a chemical leak becomes dangerous, and tune the next round of the query based

on the response obtained. We present one such adaptation function; it is conceptually simple and yet shows a high degree of success during adaptation. In this section, we focus on queries that obtain the approximate average value of the network, as it is the most typical summary statistic used in long term monitoring applications. Our adaptation function uses the confidence intervals of the average to change the retransmission probability in the next round if necessary. Confidence intervals, are often used to signify the likely range of an estimated value based on some samples from the data. The standard formula for computing the confidence length:

$$\text{ConfidenceLength}(CL) = 1.96 * \sigma / \sqrt{(n)}$$

σ is the standard deviation of the data, and n is the number of samples. The constant 1.96 indicates 95% confidence in the computed estimate.

Confidence lengths can be calculated easily but are an unintuitive way to express user preferences. It is easier for an application to express the extent of error it is willing to tolerate. We call this the *Tolerable Error*:

$$\text{TolerableError}(TE) = 100 * CL / \mu$$

TE can be easily expressed by the application as a single value. For example, a value of 10% indicates that the confidence length computed from a query response should be no more than 10% of the mean of the samples. A confidence length is small only when a large number of nodes participate or when the data is highly correlated (i.e., the standard deviation is small). Consequently, the error for a query round will be small for the same reasons. We now show how to use a simple adaptation function that employs this *Tolerable Error* to perform adaptive approximate query processing for persistent queries. Fig. 9 updates our query processing algorithm to of expanded state. First, the node stores the application-specified *Tolerable Error* (TE) for each persistent query. The state variable p becomes a list of p values, one for each round of the persistent query. They are indexed by i , the number of the round with which the particular p is associated. We also introduce a timer, *queryTimer*, which fires when it is time to issue a new round of the persistent query. It is at this time that the results for the previous round are delivered to the application. Finally, the variable *replyList* stores the samples constituting a round until the round is complete. The protocol shown in Fig. 9 is a simplified version of the actual implementation. We check the query-id before processing a node's reply packet to ensure that all responses belong to the same round.

The figure shows the replacement behavior for the QUERYREPLYRECEIVED action. Instead of immediately forwarding replies to the application, our protocol stores them in the *replyList*, and waits to aggregate the replies for the application before the next query round. We add the action SENDPERSISTENT-QUERYROUND to the formalization. This action is timer driven; when the timer expires indicating it is time to send the next one-shot query for this persistent query, this behavior is enabled. It first computes the average and the error for the samples received in the previous round and sends the result to the application. It then compares the error to the application-specified tolerable error TE .


```

QUERYREPLYRECEIVED(r)
  if r.destination = A then
    replyList := replyList ∪ r
  else
    send QueryReply(r) to parent
  end

SENDPERSISTENTQUERYROUND()
Precondition:
  queryTimer expires
Effect:
  average := computeAverage(replyList)
  error := computeError(replyList)
  /*** send average and error to application ***/
  diff := TE - error
  if |diff| < 1 then
    increment := 0.05
  else
    increment := 0.20
  end
  if diff > 0 then
    increment := -increment
  end
  pi+1 := pi + increment
  if pi+1 > 1 then
    pi+1 = 1
  end
  if pi+1 < 0.1 then
    pi+1 = 0.1
  end
  reset queryTimer

```

Fig. 9. Updated Query Processing Algorithm

Our example protocol uses the TE very simply. If the error is close the tolerable error, the protocol makes only a small adjustment in the value of p (an adjustment with a magnitude of 0.05). Otherwise the protocol makes a bigger step (an adjustment with a magnitude of 0.20). A more sophisticated adaptation would use a continuous adjustment scale, where the magnitude of the increment is computed relative to the magnitude of the TE directly. The increment is adjusted based on whether p should be raised or lowered, and then p_{i+1} is calculated. Finally, the value of p is adjusted if it went outside the range of $0 - 1$. This is a simplified example that matches what was used in our experiments. Other adaptation algorithms can be designed that use a larger history (more than just the error in the last round) so that the changes are not as abrupt. Our goal was to demonstrate the efficacy of the technique even when using a relatively simple adaptation function. In the next section, we show that even this simple

adaptation protocol is quite capable of dynamically trading message overhead for desired accuracy.

6 Evaluation

In this section, we evaluate the performance of the simple adaptation mechanism outlined in Section 5. This provides an example of how a combination of a parameterizable protocol and data quality metrics can generate a protocol that incurs the least amount of overhead possible while still satisfying application-defined requirements. We assume the application is sampling a field of sensors all measuring the same thing (e.g., the temperature of animals in a farm). The application expects the values to be similar; therefore the deviation of the results from a mean is a reasonable adaptation point. The application provides a Tolerable Error (as described in the previous section), and the protocol adapts p to dynamically target this Tolerable Error. We use the same experimental setup as outlined in Section 3. Once again, we have three types of data sets representing data with correlation varying from high to low. We provide 95% confidence intervals for our results. A persistent query is run for 300 seconds, and a new query round is issued every 25 seconds.

Fig. 10 shows the number of nodes responding to the query as the required Tolerable Error is increased. A low value of tolerable error indicates an application that requires a high degree of accuracy. Conversely, a high Tolerable Error indicates that the application does not require high fidelity data. When the Tolerable Error is very low (left of the graph), a large number of nodes need to be involved to satisfy this requirement.

When the requirement is less restrictive, receiving results from far fewer nodes will suffice. In our experiments, p is set to 0.5 during the first round. As the rounds progress, the adaptation function enforces a change in p based on the computed error. If the error is low, p is progressively increased; if it is high, p is decreased. The average value of p for the persistent query varies from 0.92 (1% TE) to about 0.16 (20% TE). Fig. 10 clearly shows that our protocol changes the number of nodes involved (and hence the communication overhead) progressively while taking into account application constraints regardless of the nature of the underlying data. However, it also shows that the

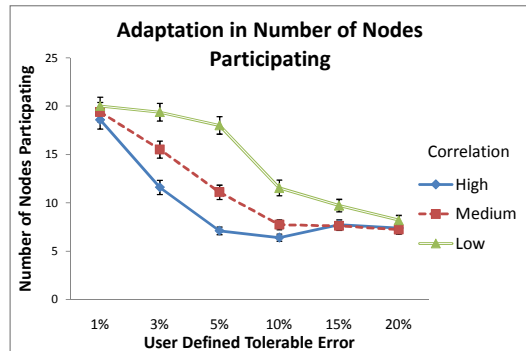


Fig. 10. Adaptation changing the number of nodes in a Persistent Protocol

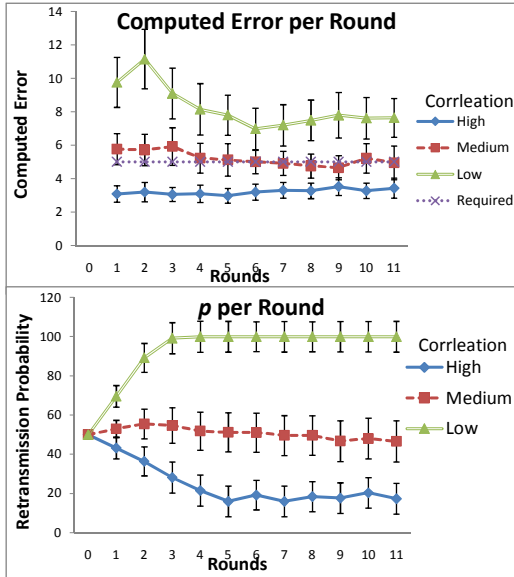


Fig. 11. Adaptation per Round

underlying data can impact the number of nodes required to successfully match user expectations. In most cases, the number of nodes required to get the same result quality when the data is loosely correlated is much more than when the data is highly. When the data is loosely correlated, the standard deviation is high, resulting in a large computed error value.

Fig. 11 takes a deeper look at the adaptation process when the application specifies that the Tolerable Error is 5%. The X axis shows the round number of the persistent query. The Y axis in the top figure shows the computed error at each round. The bottom figure shows the retransmission probability (expressed as a percentage) used by the query for each round. The first round’s one-shot query is issued with a probability of 50%. When the data is not correlated, the application is unable to meet the required error of 5% (dashed line in the top graph), and consequently ends up increasing its retransmission probability to 100% very quickly. Even when p is set to 1, the protocol is still unable to attain the user required error, but it does manage to reduce the error from 11% to 8%. The same algorithm behaves differently when the data is moderately correlated. It is relatively close to the desired Tolerable Error at about 50% retransmission probability. Consequently, it reduces its retransmission probability in small amounts until it reaches the desired degree of data quality. Once it satisfies the application’s accuracy requirements, it hovers around that value for the remaining rounds. Given highly correlated data, our adaptive protocol can achieve the application’s Tolerable Error easily with the starting value of p set to 50%. Consequently, it tries to minimize the number of nodes involved in query processing. As can be seen from the top graph, the computed error increases slightly as the

rounds progress but remains well below the Tolerable Error. The bottom graph shows that the retransmission probability drops drastically from 50% to about 17% by the end of 12 rounds. This translates to only about six nodes being involved in the query. Thus, the protocol has adaptively traded a slight loss in accuracy for a large savings in communication overhead because the accuracy loss was well within application tolerable levels.

The data used in our experiments is not jointly Gaussian. In spite of that, confidence intervals have proven to be a good point of adaptation. This suggests that gossip routing can be successfully parametrized for a large number of data distributions. From these results it can be seen that there is a large benefit to be gained from dynamically adapting the behavior of a persistent protocol based on results gathered in the previous rounds. Allowing the application to specify accuracy constraints and using that as a benchmark to adapt a protocol's behavior dynamically can lead to an ideal number of nodes answering a query. This helps answer the query effectively within the bounds of application tolerability and reduce communication overhead.

7 Related Work

Our work is broadly related to three classes of systems that exist in the literature.

Approximate Query Processing: Since performing in-network aggregation [14] by distributing the computation through out the network can be quite expensive, approximate querying techniques were designed to provide estimates of answers. CAG [20] creates clusters where nodes with highly correlated data form a group, and only the cluster head is involved in transmitting data. CAG's emphasis is network structure maintenance while ours is to adapt the approximation technique based on the dynamics of the network. Also, we avoid the overhead associated with maintaining grouping mechanisms like trees or clusters. Other approximate query processing algorithms [4,7] create models of data at the base station, and the querier interacts with the base station. The base station maintains estimates of the data at the sensor nodes and employs different techniques to keep its estimate accurate with the actual data. Our approach queries actual data at query time and also does not require any state maintenance mechanism to compute the estimates. In addition, we expose data quality metrics to add context to a query response. Finally, Backasting [18] is a technique where adaptive sampling is used to perform estimation of a spatial field and identify interesting objects, e.g., the boundary of a physical space. Adaptation is used to determine if a region is of interest by sampling a few nodes initially and then imposing a hierarchy. We focus on using adaptation over an extended period of time for persistent queries and impose no hierarchy.

Gossip Routing: We chose a gossip routing based protocol because it naturally lends itself to selectively sampling data from nodes. There has been extensive research in using the concept of gossiping for a variety of tasks. Several researchers

have incorporated gossip routing in the sensor networks domain [3,5]. Their focus is typically on studying the coverage of gossip routing for different network topologies. In contrast, we focus on using gossip routing as a mechanism to perform adaptive approximate query processing. One interesting variant is where nodes update the probability of retransmission based on the relationship between nodes in the network hierarchy [11]. Nodes are inferred to be organized as parents, children or siblings and these relationships are used to tune the retransmission probabilities. This is complementary to our work and can be used in conjunction to adapt our protocol to network topology and data distribution simultaneously. Gossip routing has also been used to perform distributed aggregate computation [10] by making nodes gossip which leads to an eventual convergence on a common value. The convergence rates of these algorithms is typically pretty slow.

Query Consistency: There has been some recent work in assessing the validity of a query response highlighting how network disruptions can render the answer to a query completely arbitrary [2,8,15]. Most of these systems use validity metrics to give an idea of the correctness of the response in the presence of node failure. Our data quality metrics provide relatively cheap context along with a query response *and* use this context directly for automatic adaptation in persistent queries.

8 Conclusion and Future Work

We presented a simple yet effective protocol to perform approximate query processing by leveraging gossip routing. We exposed meta-data in the form of quality metrics and demonstrated how they add context to a query response in both one-shot and persistent queries. Finally, we provided a protocol that uses the data quality metrics to automatically adapt approximate query processing for persistent queries. Our results demonstrate that we can effectively trade off user defined accuracy for overhead. In future, we plan to run our protocol on a real sensor network deployments and plan to incorporate temporal approximation. In addition we plan to incorporate the benefits of geographic scoping as shown in [5,16] into our adaptive algorithm.

References

1. Bash, B., Byers, J., Considine, J.: Approximately uniform random sampling in sensor networks. In: Proc. of Wkshp on DMSN, pp. 32–39 (2004)
2. Bawa, M., Gionis, A., Garcia-Molina, H., Motwani, R.: The price of validity in dynamic networks. Proc. of ACM SIGMOD 73(3), 245–264 (2007)
3. Braginsky, D., Estrin, D.: Rumor routing algorithm for sensor networks. In: Proc. of WSNA, pp. 22–31 (2002)
4. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-driven data acquisition in sensor networks. In: Proc. of VLDB, pp. 588–599 (2004)

5. Dimakis, A.G., Sarwate, A.D., Wainwright, M.J.: Geographic gossip: efficient aggregation for sensor networks. In: Proc. of IPSN, pp. 69–76 (2006)
6. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.S.J.: System architecture directions for networked sensors. In: Proc. of ASPLOS, pp. 93–104 (2000)
7. Jain, A., Chang, E.Y.: Adaptive sampling for sensor networks. In: Proc. of DMSN, pp. 10–16 (2004)
8. Jain, N., Kit, D., Yalagandula, D.M.P., Dahlin, M., Zhang, Y.: Network imprecision: A new consistency metric for scalable monitoring. In: Proc. of OSDI (2008)
9. Jindal, A., Psounis, K.: Modeling spatially correlated data in sensor networks. ACM TOSN 2(4), 466–499 (2006)
10. Kempe, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: Proc. of IEEE FOCS, pp. 482–491 (2003)
11. Kyasanur, P., Choudhury, R.R., Gupta, I.: Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. In: Proc. of MASS, pp. 91–100 (2006)
12. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: accurate and scalable simulation tinyos applications. In: Proc. of IEEE SenSys, pp. 126–137 (2003)
13. Lynch, N., Tuttle, M.: An introduction to I/O automata. In: CWI-Quarterly, pp. 219–246 (1989)
14. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: TAG: A tiny aggregation service for ad hoc sensor networks. In: Proc. of OSDI, pp. 131–146 (2002)
15. Payton, J., Julien, C., Roman, G.-C.: Automatic consistency assessment for query results in dynamic environments. In: Proc. of ESEC/FSE, pp. 245–254 (2007)
16. Sarkar, R., Zhu, X., Gao, J.: Hierarchical spatial gossip for multi-resolution representations in sensor networks. In: Proc. of IPSN, pp. 420–429 (2007)
17. Skordylis, A., Trigoni, N., Guitton, A.: A study of approximate data management techniques for sensor networks. In: Proc. of WISE, June 2006, pp. 1–12 (2006)
18. Willett, R., Martin, A., Nowak, R.: Backcasting: adaptive sampling for sensor networks. In: Proc. of IPSN, New York, NY, USA, pp. 124–133 (2004)
19. Xu, N.: A survey of sensor network applications. Technical report, The University of Southern California (2002)
20. Yoon, S., Shahabi, C.: The clustered aggregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. ACM TOSN 3(1), 3 (2007)

On-Demand Node Reclamation and Replacement for Guaranteed Area Coverage in Long-Lived Sensor Networks

Bin Tong¹, Zi Li¹, Guiling Wang², and Wensheng Zhang¹

¹ Department of Computer Science, Iowa State University

² Department of Computer Science, New Jersey Institute of Technology
{tongbin, zili, wzhang}@cs.iastate.edu, gwang@njit.edu

Abstract. To achieve required sensing coverage for a very long period of time is an important and challenging problem in sensor network design. Recently, Tong et al. have proposed a *node replacement and reclamation (NRR) strategy*, and designed an *adaptive rendezvous-based two-tier scheduling (ARTS) scheme*. However, the ARTS scheme only considers point coverage but not area coverage, which is required in many applications. To address this limitation, we propose in this paper a new implementing scheme for the NRR strategy based on a novel *staircase-based scheduling model*. Extensive simulations have been conducted to verify that the proposed scheme is effective and efficient.

Keywords: Sensor Networks, Reclamation and Replacement, Duty-cycle Scheduling.

1 Introduction

In a wireless sensor network (WSN), sensor nodes are powered by batteries that can operate for only a short period of time, resulting in limited network lifetime if batteries are not replaced. The limited lifetime may disable its application in long-term tasks such as structural health monitoring for bridges and tunnels, border surveillance, road condition monitoring, and so on. Hence, many energy conservation schemes [1] were proposed to battle the constraint. These schemes can slow down the rate of energy consumption, but cannot compensate energy consumed. Fully addressing the problem requires energy to be replenishable to sensor nodes. One approach is to harvest energy from various environmental sources [2, 3, 4, 5, 6] such as the sunlight. The amount of energy that a solar cell can harvest is proportional to its surface area, but it is infeasible to equip a tiny sensor node with a large-size solar cell. The amount of available solar energy also depends on uncontrollable conditions such as cloudiness of the sky. Therefore, it is likely that the energy harvested is limited and unable to satisfy the needs of sensor nodes. Another approach is to incrementally deploy new sensor nodes to take over sensor nodes running out of energy. However, this approach is costly because sensor node hardware cannot be reused, and more importantly, it causes pollution to the environment because dead batteries and hardware are left in the environment. Therefore, seeking an effective and efficient way to guarantee long-term energy supply has persisted as a big challenge.

Recently, Tong *et al.* [7] proposed a *node reclamation and replacement (NRR)* strategy. With this strategy, a robot or human labor called *mobile repairman (MR)* periodically reclaims sensor nodes of low or no energy supply, replaces them with fully-charged ones, and brings the reclaimed sensor nodes back to a place called *energy station (ES)* for recharging. An *adaptive rendezvous-based two-tier scheduling (ARTS) scheme* [7] has also been designed to realize the NRR strategy. However, the ARTS scheme only considers point coverage [8,9]. That is, it is assumed that sensor nodes are deployed to monitor points of interest scattered in a network field, while in many application scenarios such as border surveillance, guaranteeing area coverage [10,11,12] is desired.

To address the limitation of the ARTS scheme, we propose in this paper another implementing scheme of the NRR strategy to achieve guaranteed area coverage in long-lived sensor networks. Our proposed scheme consists of three tightly coupled components: (i) the protocol for sensors to coordinate their duty-cycle scheduling locally, (ii) the protocol for sensors and the ES to communicate with each other, and (iii) the algorithm for the ES to determine how to perform node reclamation and replacement on demand. These three components work together to achieve the following objectives: (a) required area coverage is guaranteed without disruption in the field monitored by the sensor network; (b) energy is replenished to the sensor network in an on-demand fashion to ensure infinite lifetime of the network and energy efficiency. The major contributions of this paper are as follows:

- To the best of our knowledge, this is the first effort that defines and addresses the problem of ensuring area coverage for an infinite period of time in sensor networks, under the node reclamation and replacement framework.
- We design a novel *staircase-based scheduling model* to address the important and challenging problem of achieving infinite network lifetime with limited number of backup sensor nodes. We have also found interesting results such as the minimum/maximum number of backup nodes that are needed to achieve infinite network lifetime. These results can help users of the proposed scheme to choose appropriate system parameters.
- Extensive simulations have been conducted to verify the effectiveness and efficiency of the scheme, as well as the validity of the findings we have discovered in theory.

The rest of the paper is organized as follows: Section 2 describes the system model. An overview of the proposed scheme is presented in Section 3, which is followed by the detailed description in Section 4. Section 5 discusses some fundamental and practical issues. Section 6 reports simulation results. Section 7 summarizes related work, and finally Section 8 concludes the paper.

2 System Model

We consider a network of n sensors, denoted as $s_1, s_2, s_3, \dots, s_n$, is deployed to a continuous field for long-term monitoring. The monitored field is divided into m small areas, denoted as $a_1, a_2, a_3, \dots, a_m$, such that, within any area a_i , the required sensing coverage level is the same at any point of the area.

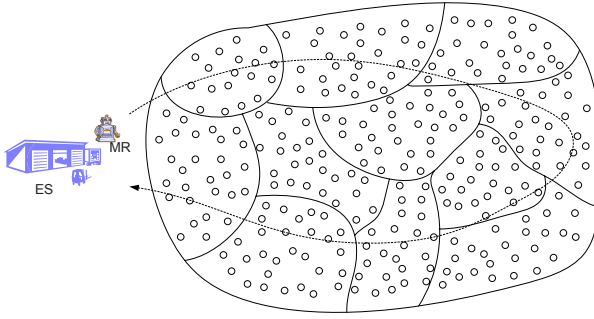


Fig. 1. System architecture

As shown in Fig. 1 the whole NRR system is composed of an *energy station (ES)*, a *mobile repairman (MR)*, and a sensor network. The ES stores a certain number (denoted as x) of backup sensors, and can recharge energy to sensors. The MR can be a human technician or a mobile robot. The MR can traverse the sensor network, reclaiming sensors of no or low energy, replacing them with fully-charged ones, and bringing the reclaimed ones back to the ES for recharging. Other assumptions of the system are as follows:

- All sensors are time synchronized. Time is divided into phases. A phase is a basic scheduling unit for duty-cycle scheduling; i.e., a sensor will not change its mode (active or sleeping) during a phase.
- A sensor has two modes: *active* and *sleeping*. For every phase, if a sensor is in the active mode, its energy is reduced by a fixed amount; if it is in the sleeping mode, its energy is unchanged. Let the energy of a fully-charged sensor be e . If a sensor is in the active mode all the time, its lifetime is denoted as T .
- For each area, the required sensing coverage level varies from N_{min} to N_{max} , subject to certain (e.g., Gaussian) distribution.
- Each area is deployed with $N_{max} + N_{back}$ (N_{back} is an integer greater than or equal to 1) *disjoint* sets of sensors, where each set of sensors can completely cover the area. That is, every point in the area can be covered by at least one sensor in each of the sets. We call these sets *coverage sets*. The reason for having more than N_{max} sets of sensors is to avoid service disruption at the time of node reclamation and replacement (Note: node reclamation and replacement cannot be completed in non-negligible time; hence, reclamation and replacement will inevitably disrupt the working of nodes that are reclaimed or newly placed).
- The MR has orientation and localization ability such that it can travel to designated locales and perform sensor replacement task. In this paper, we assume that the MR is able to carry x sensors a time. This can be relaxed to the case that the capacity of MR is smaller, and the trip scheduling algorithm studied in [7] may be applied to address this problem.
- Charging a sensor at the ES takes *non-negligible* time, which is denoted as τ . Note that, sensors can be recharged in parallel, we assume that it is possible to recharge all x backup sensors managed by the ES at the same time.

Design Goal. In this paper, we aim to design a collaborative scheduling scheme for sensors and the node reclamation and replacement algorithm for the ES/MR, such that (i) the sensor network can maintain the required area coverage for an infinite period of time, and (ii) the number of travels the MR should take is as small as possible (i.e., the average interval between two consecutive replacement trips is as large as possible).

3 Overview of the Proposed Scheme

3.1 Key Ideas

To achieve guaranteed area coverage for an infinite period of time, two necessary tasks should be performed: firstly, sensors should collaboratively schedule their duty-cycles to achieve required area coverage; secondly, sensors and the ES/MR should coordinate to replenish energy into the network through node reclamation and replacement.

If the ES have unlimited number of backup sensors to use and the reclamation/replacement can be finished instantly, the above two tasks can be achieved easily. For example, any existing collaboratively duty-cycle scheduling schemes [13] can be applied for the first task; as for the second task, whenever an area is short of alive sensors, a request is sent to the ES, which then dispatches the MR to reclaim and replace sensors for the area. In reality, however, the backup sensors owned by the ES are limited and should be not too large for economic reasons, and the recharging take non-negligible time. Using the above naive approach, it may happen that, at some time instance, 1000 sensors should be replaced while the ES has only 500 backup sensors.

To address the above problem, the duty-cycle scheduling of sensors and the node reclamation and replacement activities should be carefully planned. In our design, we propose a *staircase scheduling model* for this purpose. The key ideas are as follows:

Coverage Set-level Scheduling. In each area, sensors are grouped into disjoint coverage sets, where nodes in each single coverage set can together cover any points in the area. Sensors are scheduled in the unit of coverage sets.

Intra-group Staircase. In each area, coverage sets are scheduled in a thoughtful way that, the required area coverage is guaranteed and meanwhile, the remaining energy levels of different sets are kept different, which form a *staircase* among the sets. Hence, different sets can be reclaimed and replaced at different time instances. As to be elaborated later, this facilitates the ES/MR to temporally reuse limited number of backup nodes to maintain lifetime.

Inter-group Staircase. Intra-group staircase may not be sufficient. It is likely that each of multiple areas needs to replace one of their coverage sets at the same time instance, and the demanded number of backup sensors could exceed what can be offered by the ES. To avoid this inter-group congestion of demands, our delicately designed scheduling strategy ensures that different areas issue demands at different time instances. This way, inter-group staircase is formed to further scatter demands and thus provide more flexibility to the ES/MR to plan the reclamation and replacement activities.

Redundancy for Flexibility. If the replacement requests issued by every area should be satisfied immediately by the ES/MR, the flexibility for performing reclamation and replacement activities will be strictly limited. At least, the number of trips taken by the MR may be too large, which may incur high system maintenance overhead. To address this issue, redundant nodes are deployed to areas to form backup coverage sets. With these backup sets, replacement requests can be satisfied with some delay, which allows the ES/MR to use one trip to satisfy multiple requests to reduce the maintenance cost.

3.2 Framework

Based on the above key ideas, the framework of our scheme is summarized as follows:

Duty-Cycle Scheduling. In our scheme, sensors in each area a_i are grouped into $N_{max} + N_{back}$ disjoint coverage sets, denoted as $cs_1, cs_2, \dots, cs_{(N_{max}+N_{back})}$, where nodes in each coverage set can together sense every point in the area. The sensors in the same coverage set are scheduled together as an integral entity. Hence, sensors in the same coverage set have similar remaining energy at any time; to simplify scheduling, we assume all sensors in the same coverage set have the same remaining energy level. All coverage sets fall into two categories: N_{max} primary sets and N_{back} backup sets. At any phase, only primary sets can be scheduled, and a coverage set can change its role from primary to backup and vice versa. Each sensor knows which coverage set it belongs to, and also maintains the information of the remaining energy levels of sensors in other coverage sets. Therefore, every sensor in each area has a consistent view regarding the remaining energy levels of sensors in the same area.

In each area, a *head* is elected among all sensors through a certain collaborative selection algorithm [14], and the role is rotated among the nodes to balance energy consumption. At the beginning of each phase, the head broadcasts the coverage requirement for the current phase, i.e., the number of coverage sets (called *coverage number*) that shall be active. How to determine the coverage number is application-dependent and out of the scope of this paper. A possible approach is, the coverage number is determined based on the observations by active sensors in the last phase; if some event was detected in the last phase, the coverage number may be increased and vice versa. At the beginning of a phase, all sensors will wake up and listen to the broadcast of the coverage number. Upon receipt of the coverage number, each sensor runs our proposed duty-cycle scheduling algorithm independently to determine whether it should be active or not. Since all sensors in an area have the consistent view about the remaining energy level of all nodes in the same area, they will arrive at the same scheduling decision.

Interactions between Area Heads and the ES. Our duty-cycle scheduling algorithm ensures that, different primary sets will use up their energy at different time instances. Shortly before a primary set (say, cs_i) of sensors uses up its energy, it hands over its duty to a backup set (say, cs_j), which has full energy. After the handoff, cs_i becomes a backup set waiting to be reclaimed and replaced, while cs_j becomes a primary set. Meanwhile, the head of the area sends a *ready* message to the ES with the number of sensors in cs_i , which is the number of sensors that need to be reclaimed and replaced. Specifically, the ready message has the following format:

$$ready\langle a, cs_i, c \rangle,$$

where a is the ID of the area, cs_i is the ID of the coverage set needing to be reclaimed and replaced, and c is the total number of sensors in the cs_i .

If a primary set is about to use up its energy, and there is no backup set with fully-charged nodes to which the primary set can hand over its duty to, the head of the area sends out a *deadline* message to the ES. Specifically, the deadline message has the following format:

$$deadline\langle a \rangle,$$

where a is the ID of the area.

Node Reclamation and Replacement. Alg. 1 formally describes how the ES responds to the above ready and deadline messages. Specifically, when the ES receives a ready message, it accumulates the total number of sensors that are ready to be replaced. The ES will dispatch the MR when either of the following conditions is true: (i) It receives a deadline message; or (ii) the total number of sensors that are ready to be replaced exceeds x .

Algorithm 1. Reclamation and Replacement Scheduling: for the ES

Notations:

- x : number of backup sensors
- R : set of ready messages that have not been served
- t : total number of sensors that are ready to be replaced

Initialization:

- 1: $R \leftarrow \phi$
- 2: $t \leftarrow 0$

Upon receipt of a ready message: $ready\langle a, cs, c \rangle$

- 3: $R \leftarrow R \cup ready$
- 4: $t \leftarrow t + c$
- 5: **if** $t \geq x$ **then**
- 6: Dispatch the MR to serve the earliest x replacement requests.
- 7: $t \leftarrow t - x$
- 8: $R \leftarrow R - \{\text{served requests}\}$

Upon receipt of a deadline message: $deadline\langle a \rangle$

- 9: Dispatch the MR to serve all pending replacement requests
 - 10: $R \leftarrow \phi$
 - 11: $t \leftarrow 0$
-

4 Detailed Description of the Scheme

The duty-cycle scheduling scheme is performed at each sensor in each area at the beginning of each phase. The input to the duty-cycle scheduling scheme is (i) the estimated remaining energy level of every sensor in all coverage sets and (ii) the coverage number for the current phase. The output of the scheme is the coverage sets that should be active in the current phase. To ease understanding, we first describe how the scheduling scheme works when the coverage number of every area is fixed (i.e., N_{max}), which is followed by the general case where the coverage number of every area is variable ranging from N_{min} to N_{max} .

4.1 A Special Case: Fixed Coverage Requirement

Suppose for each area a_i , the number of sensors in each coverage set of area a_i , $1 \leq i \leq m$, is denoted as c_i . Since areas are divided based on coverage requirement, c_i could be different for different areas. For each area a_i , $1 \leq i \leq m$, we need to schedule all N_{max} primary coverage sets at any phase.

For all the N_{max} primary coverage sets, we let their remaining energy per node form a “staircase”, and the height of each stair is

$$\frac{e}{N_{max}},$$

where e is the amount of full energy of a sensor. The formation procedure of this staircase is discussed later.

Fig. 2 shows an example where the monitored field consists of four areas. Each row in Fig. 2 shows the snapshot of remaining energy of each coverage set in each area at different time points. As can be seen, out of five coverage sets in each area, one is in the backup role, and the other four are in the primary role. The remaining energy per node of the four primary coverage sets forms a staircase with a stair height of $e/4$.

In our scheme, we define an order in which areas are visited by the MR to reclaim and replace sensors in these areas. For any two areas that are to be visited consecutively, their staircases have a phase difference δ , where δ and the height of a stair have the following relation:

$$\frac{e}{N_{max}} = m\delta, \quad (1)$$

where m is the number of areas. In Fig. 2, areas are sorted as a_1, a_2, a_3, a_4 . As can be seen, at time point 0, the staircase of primary coverage sets in a_2 is $e/16$ higher than that of the primary coverage sets in a_1 , the staircase of the primary coverage sets in a_3 is also $e/16$ higher than that of the primary coverage sets in a_2 , and so on. This phase difference remains as time evolves.

Since the coverage requirement is always N_{max} , all the four primary coverage sets will be active at any time. When the primary coverage set with the minimum energy drains of its energy, it will (i) shift its duty to a backup coverage set, which has full energy; (ii) becomes a backup set. Meanwhile, the head of the area will send a ready message to the ES, and the full energy backup coverage set will become a primary coverage set.

In Fig. 2, at time $t = T/16$, the primary coverage set with the minimum energy in a_1 drains of its energy, and shifts its duty to the only backup coverage set. A ready message is also sent to the ES. Since at this time, the total number of nodes that are ready to be replaced is 16, which is less than $x = 32$, the MR will wait. At time $t = T/8$, the primary coverage set with the minimum energy in a_2 drains of its energy, and shifts its duty to the backup coverage set. A ready message is also sent to the ES. At this time, the total number of nodes that are ready to be replaced equals to x . Thus, the MR makes a replacement tour, replacing nodes in the backup sets of a_1 and a_2 . Similarly, the MR makes another replacement tour at $t = T/4$, replacing nodes in the backup coverage sets of a_3 and a_4 .

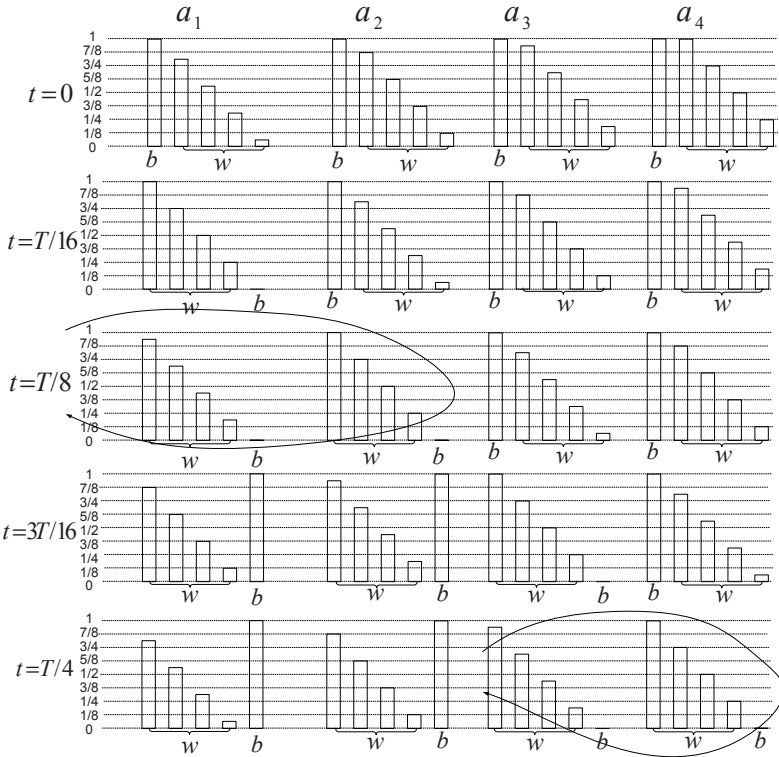


Fig. 2. Example 1: duty-cycle scheduling. Each bar represents a coverage set. $N_{max} = 4$, $N_{back} = 1$, $m = 4$, and $x = 32$. Each coverage set in every area has 16 sensors. “w” means primary set, and “b” means backup set.

One noteworthy fact is that, in this example, recharging x sensors should be completed in $T/8$. We have derived a relation between recharging time and the minimum number of backup sensors needed, which is to be discussed later.

Staircase Formation. In the above, we assume that the staircase structure is already formed. However, when a sensor network starts operating, all sensors in the sensing field have full energy. To form the staircase structure, we propose the following method. Without loss of generality, we assume the pre-defined visiting order to the areas is $\langle a_1, a_2, \dots, a_m \rangle$. When a primary coverage set in a_1 consumes δ energy¹, it shifts its duty to a backup coverage set, and becomes a backup coverage set itself. The head of area a_1 also sends a ready message to the ES. Similarly, when a primary coverage set in a_2 consumes 2δ energy, it shifts its duty to a backup coverage set, and becomes a backup coverage set itself. Besides, the head of area a_2 sends a ready message to the ES. In general, a primary coverage set in a_i will make the role transition and trigger ready message reporting after it consumes $i\delta$ energy.

¹ Since all primary coverage sets will have the same remaining energy at that time, we randomly pick one.

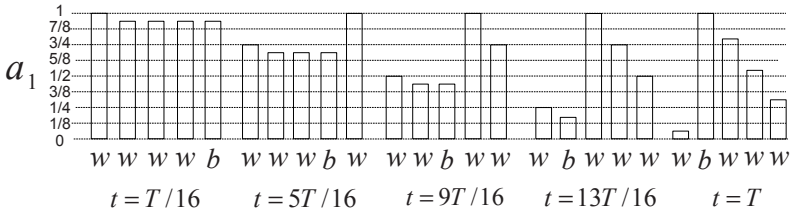


Fig. 3. Example 2: initial staircase formation of area a_1 in Fig. 2 $\delta = T/16$

The next time for role transition and ready message reporting in a_1 is after a primary coverage set with the minimum energy has consumed $m\delta$ energy after the first role transition. The third time for role transition and ready message reporting in a_1 is after a primary coverage set with the minimum energy has consumed $m\delta$ energy after the second role transition; and so on. Other areas will follow the same rule to conduct their role transitions and ready message reporting. After time T , the staircase structure will be naturally formed. Fig. 3 shows an example of staircase formation of area a_1 in Fig. 2. Note that, the staircase shown at $(t = T)$ is the same as that at $(t = 0)$ in Fig. 2.

4.2 General Case: Variable Coverage Requirement

In this section, we consider the general case that the required coverage number is not always N_{max} , but varies in range $[N_{min}, N_{max}]$.

Given an area $a_i, 1 \leq i \leq m$, we let the remaining energy per node of its N_{max} primary coverage sets, denoted as $w_1, w_2, \dots, w_{N_{max}}$, form a staircase as described above. Assume $e_i, 1 \leq i \leq N_{max}$ represents the remaining energy of coverage set w_i . Without loss of generality, we have $e_1 < e_2 < e_3 < \dots < e_{N_{max}}$, where the difference between any two consecutive terms is $m\delta$. The duty-cycle scheduling is performed phase by phase.

Assuming the coverage number for the first phase is $q_0, N_{min} \leq q_0 \leq N_{max}$, we will need to schedule q_0 primary coverage sets. In our scheme, we schedule primary coverage sets $\{w_1, w_2, w_3, \dots, w_{q_0}\}$ for the first phase. If the coverage number for the next phase is $q_1, N_{min} \leq q_1 \leq N_{max}$, we will need to schedule q_1 primary coverage sets. In this case, we will schedule coverage sets

$$w_{(q_0+1) \bmod N_{max}}, w_{(q_0+2) \bmod N_{max}}, \dots, w_{(q_0+q_1) \bmod N_{max}}$$

In other words, we adopt a round-robin scheduling policy while maintaining the staircase structure.

In this case, whenever each area a_i uses up its primary coverage set with the minimum energy, its head sends a ready message to the ES if there are backup coverage sets with full energy. If all the backup coverage sets have empty energy before the primary coverage set with the minimum energy is about to use up its energy, the head will send a deadline message to the ES.

The formal duty-cycle scheduling algorithm for the variable coverage number case is described in Alg. 2.

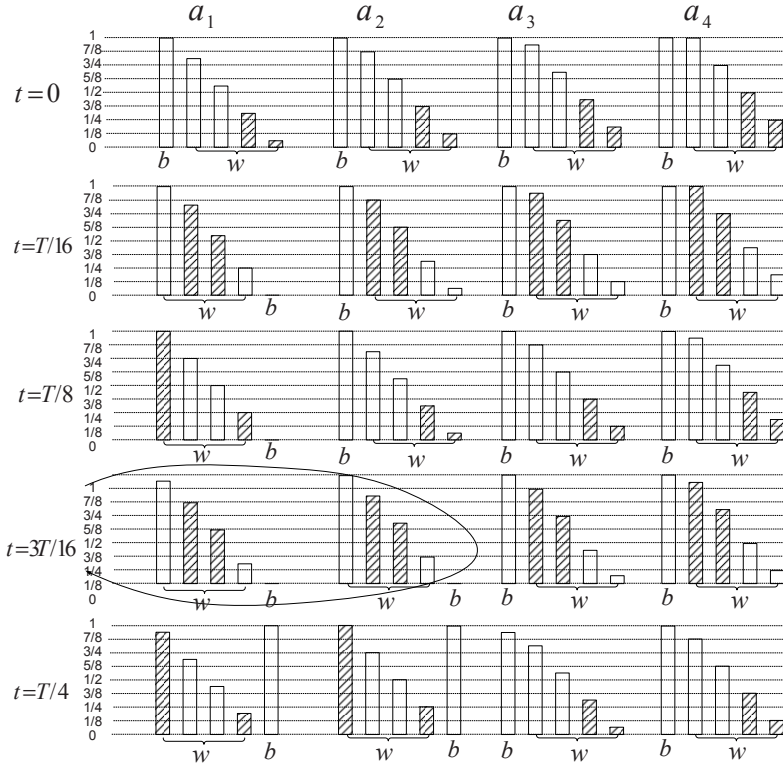


Fig. 4. Example 3: duty-cycle scheduling. Each bar represents a coverage set. Shaded bars are scheduled in the current phase. $N_{max} = 4$, $N_{back} = 1$, $m = 4$, and $x = 32$. Each coverage set in every area has 16 sensors. Phase length is $T/16$.

Fig. 4 shows an example. In this example, the length of a phase is $T/16$, and the coverage number is two for the first four phases for all areas. As can be seen, in the first phase, the two primary coverage sets with the minimum remaining energy in all areas are scheduled. In the second phase, the next two primary coverage sets in all areas are scheduled. This process is carried on.

At time $t = T/16$, the head of area a_1 sends out a ready message to the ES. The MR will not make a replacement tour since the number of sensors that are ready to be replaced, 16, is less than $x = 32$. At time $t = 3T/16$, the head of area a_2 sends another ready message. At this time, the number of sensors that are ready to be replaced reaches x , and thus the MR conducts a replacement.

Staircase Formation. The staircase formation procedure for the variable coverage number case is the same as that for the fixed coverage requirement case.

Algorithm 2. Duty-Cycle Scheduling for the variable coverage requirement case: for sensors in primary coverage set $w_i, 1 \leq i \leq N_{max}$

Notations:

q : coverage number in phase p

$b_j, 1 \leq j \leq N_{back}$: N_{back} backup coverage sets

var $start$; // start position of primary coverage sets for phase p .

- 1: **if** $w_i \in \{w_{start}, w_{((start+1) \bmod N_{max})}, \dots, w_{((start+q-1) \bmod N_{max})}\}$ **then**
 - 2: Schedule coverage set w_i .
 - 3: **if** w_i drains of its energy **then**
 - 4: Randomly choose a backup coverage set with full energy, b_j .
 - 5: Coverage set w_i changes its role to backup.
 - 6: Coverage set b_j changes its role to primary.
 - 7: $start \leftarrow (start + q) \bmod N_{max}$
-

5 Discussions

5.1 Lower Bound of Required Number of Backup Nodes

Since charging batteries takes non-negligible time, the energy replenishment rate is affected by the number of backup nodes owned by the ES. Assuming the number of backup nodes is x , the time to recharge a sensor is τ , and full energy of a sensor is e , the energy replenishment rate is

$$xe/\tau$$

This rate should be large enough to compensate energy consumption of the network even in the worst case scenario. Specifically, the worst case energy consumption rate occurs when the coverage number in each area is N_{max} .

Consider area $a_i, 1 \leq i \leq m$, in which each coverage set has c_i sensors. N_{max} coverage sets will each consume e/N_{max} energy in T/N_{max} time, where T is a sensor's lifetime. Thus the total energy consumption of area a_i in T/N_{max} time is

$$c_i N_{max} \frac{e}{N_{max}} = c_i e$$

It follows that the energy consumption rate in area a_i is $c_i e N_{max}/T$.

The total energy consumption rate over all areas is

$$\frac{e}{T} \sum_{i=1}^m c_i N_{max}$$

We have

$$\begin{aligned} \frac{xe}{\tau} &\geq \frac{e}{T} N_{max} \sum_{i=1}^m c_i \\ x &\geq \frac{\tau}{T} N_{max} \sum_{i=1}^m c_i \end{aligned} \tag{2}$$

5.2 Upper Bound of Number of Backup Nodes

In the proposed scheme, the MR only replaces sensors in backup coverage sets for each area. The reason is that replacement will disrupt sensor nodes' operation. By not replacing the N_{max} primary coverage sets, service disruption is avoided.

As a result, at one time, the maximum number of sensors that are ready to be replaced in area a_i is $N_{back}c_i$, and the total number of sensors that are ready to be replaced over all areas is

$$N_{back} \sum_{i=1}^m c_i \quad (3)$$

In general, this is the upper bound for x in the sense that if $x > N_{back} \sum_{i=1}^m c_i$, the surplus backup sensors will never be used.

However, there is an exception when the lower bound calculated by Eq. (2) is greater than the upper bound calculated by Eq. (3). This case is discussed in the following.

5.3 Impact of Node Recharging Time

If sensor recharging time at the ES is very long, it is possible that the lower bound of x calculated by Eq. (2) is greater than the upper bound calculated by Eq. (3). Here we face a dilemma: On one hand, x should be greater than the calculated lower bound in order to guarantee the coverage requirement over an infinite period of time; on the other hand, if x is greater than the calculated upper bound, the surplus sensors will not be used. We propose the following method to address this issue.

Assume the lower bound of x calculated by Eq. (2) is denoted as l , and the upper bound calculated by Eq. (3) is denoted as h . Given sensor recharging time τ , we list its divisors by natural numbers $2, 3 \dots$, and for each divisor, we calculate a lower bound l' using Eq. (2). This process stops $l' < h$. Assume at this time the divisor of τ is $\tau/k, k \geq 2$.

If we have $kh \geq x \geq kl'$, then the x backup sensors can be divided into k batches. All sensors in a batch will start being recharged at the ES at the same time. Further, we order the k batches into a sequence, and the start times for any two consecutive batches in the sequence being recharged differ by τ/k . In other words, the system generates $x/k, h \geq x/k \geq l'$, fully charged sensors every τ/k . This way, the proposed scheme works as the regular case.

5.4 Some Practical Issues

Next, we discuss some practical issues in implementing the proposed scheme.

First, sensor nodes may fail at any time. Our scheme can tolerate sensor failures, i.e., failed sensors will be replaced by the MR. We employ the following method to detect sensor failures. At the time for scheduling (i.e., at the beginning of a phase), if a primary coverage set w is chosen to be active in the phase, all sensors in the coverage set will send a message to the head of the area. If the head does not receive the message from a sensor u for more than a threshold of times, it considers u has failed, and then sends a *failure* message to the ES. The MR will replace the failed sensor in its next replacement trip.

Second, our scheme requires communication between active sensors and the head of each area in every phase. Since the size of an area is typically small, the imbalance in energy consumption among sensor nodes for forwarding data packets is limited. Further, we factor the maximum energy consumption for packet forwarding into total energy consumption at each sensor.

Third, the head of each area will report ready and deadline messages, which may travel a long route. However, reporting of these messages is infrequent since they are only sent out when the area have consumed considerable amount of energy, which is on the magnitude of sensor batteries's lifetime.

6 Performance Evaluation

We built a custom simulator using C++ to evaluate the performance of the proposed scheme.

6.1 Experimental Settings, Metrics and Methodology

Table 1 shows system parameters we used in the simulation. We consider a sensor network composed of 80 areas. Each area has $(N_{max} + N_{back})$ disjoint coverage sets, and each of which is able to cover the whole area. The number of sensors in each coverage set is a random number, which complies to a Gaussian distribution, $Gau(16, 3)$, with mean of 16.

In the experiments, we normalize the full energy level of a sensor to 1440 units and the energy consumption rate is 0.1 unit/minute if the sensor is active. Thus, each sensor's lifetime T is 240 hours, i.e., 5 days. The length of a phase is set to 10 minutes. Coverage numbers for each area vary between N_{min} and N_{max} . N_{min} is set to 1, and N_{max} is set to 4 in all experiments.

In reality, coverage number is determined by the application, as well as the real-time frequency and distribution of events. In our simulation, coverage number complies to

Table 1. General experimental settings

field size	500m * 500m
# of areas	80
sensing range	20m
transmission range	40m
N_{min}	1
N_{max}	4
N_{back}	{1, 2, 3}
recharging time	6 hours
sensor's lifetime time	240 hours (5 days)
# of sensors per coverage set	$Gau(16, 3)$
sensor's full energy	1440 units
phase length	10 minutes
energy consumption rate	0.1 unit/minute
cut-off time	4800 hours (200 days)

a truncated Gaussian distribution, which is $Gau(\mu = N_{min}, \sigma = 2)$ truncated to the range $[N_{min}, N_{max}]$.

The performance metrics include:

- *Average replacement interval*: Average time between two consecutive replacement tours made by the MR.
- *Average utilization of the MR*: The MR may not carry x sensors in each replacement tour due to the replacement deadlines set by each area. Average utilization of the MR is the average ratio of the number of backup sensors actually carried by the MR to x .
- *Distribution of replacement intervals*: To ease reclamation/replacement planning, a distribution of replacement intervals with smaller variance is preferred in practice.

We consider the following sets of scenarios: (i) All areas have the same coverage number at any time, and (ii) All areas subject to the same distribution of coverage numbers, but coverage numbers in all areas are independent of each other. For each experiment, our proposed scheme is executed for a long time period, starting at 0 and ending at a *cut-off* time. The cutoff time is set to 4800 hours, i.e, 200 days, for all experiments. Furthermore, we run each simulation for 50 times for the metrics of average replacement interval and average utilization of the MR, and 500 times for the metric of distribution of replacement intervals, and take average for each of the metrics.

6.2 Scenario I: Same Coverage Number for All Areas

In this experiment, coverage number is the same for all areas at any phase. The number of backup coverage sets, N_{back} , varies among $\{1, 2, 3\}$. The results are shown in Fig. 5. Fig. 5(a) and Fig. 5(b) show the trend of average replacement interval and utilization of the MR when coverage number complies to the truncated Gaussian distribution. As can be seen, given the number of backup coverage sets, average replacement interval increases as the number of backup sensors, x , increases in an approximately linear fashion. At the same time, the utilization of the MR keeps at 1. However, when x reaches a certain value, the average replacement interval levels off, and at the same time, the utilization of the MR starts to drop.

For example, given one backup coverage set for each area, when x exceeds 1300, the average replacement interval stops increasing, and the utility of the MR drops to 0.95.

The reason for this phenomenon is explained as follows. Since all areas have the same coverage number, their primary coverage sets consume their energy at the same rate. Further, in our scheme, the remaining energy of primary coverage sets in any two consecutive areas according to the pre-defined visiting order has a phase difference δ . Therefore, the time instances for the heads in all areas to send ready messages are evenly distributed as time evolves.

When x is small, the time instances when the number of sensors that are ready to be replaced exceeds x are always ahead of arrival of any deadline message. Thus, the MR will replace x sensors in each replacement tour, which results in a full MR utilization. Further, given the total amount of energy consumption of the network until the cutoff time, the total amount of energy that is needed to be replenished into the network is fixed. As a result, average replenish interval increases with x in a linear fashion.

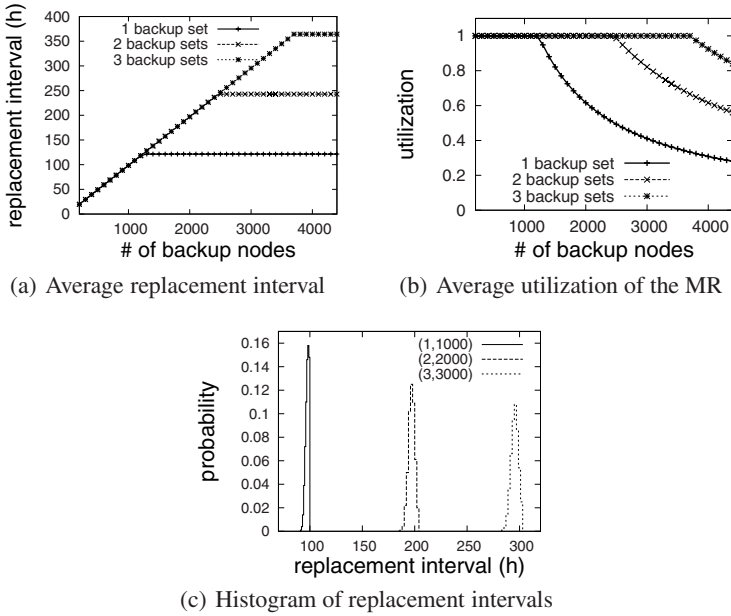


Fig. 5. Scenario I: Same Coverage Number for All Areas

On the other hand, when x exceeds the upper bound of x calculated by Eq. (3), which is between 1200 and 1300 in this experiment, a deadline message will arrive before the number of sensors that are ready to be replaced reaches x . Therefore, replacement interval stops to increase at this point. Furthermore, since the number of backup sensors that are actually used stays at the upper bound value, as x increases, the utilization of the MR decreases in a reciprocal fashion.

The results show that given a fixed number, N_{back} , of backup coverage sets, we cannot raise average replacement interval over a certain value by simple increasing x . Instead, N_{back} will need to be increased.

Fig. 5(c) shows histograms of replacement intervals for three different parameter sets. In Fig. 5(c), the first number in a pair of parentheses is the number of backup coverage sets, and the second number is x . For example, “(1,1000)” means one backup coverage set and 1000 backup sensors. Note that for all the three parameter sets, the utilization of the MR is 1. As can be seen in Fig. 5(c), replacement intervals cluster in a small range. For parameter set (1,1000), the mean of replacement intervals is 98.53, and the standard deviation is 2.35.

6.3 Scenario II: Same Coverage Number Distribution for All Areas

In this experiment, all areas have the same value of $N_{max} = 4$, and their coverage numbers comply to the same probability distribution. However, coverage numbers of different areas are independent of each other. In addition, we assume in each area, coverage numbers at different phases are independent of each other.

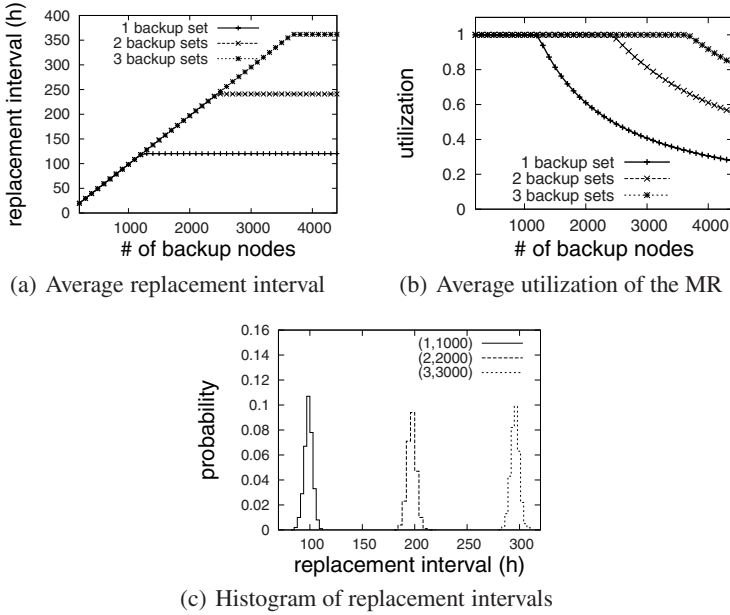


Fig. 6. Scenario II: Same Coverage Number Distribution for All Areas

Fig. 6(a) and Fig. 6(b) show very similar patterns as in Fig. 5(a) and Fig. 5(b) respectively. This can be explained as follows.

Since coverage number is a random variable between N_{min} and N_{max} , and coverage numbers at different phases are independent of each other, the summation of coverage numbers over a large number of phases can be approximated with a Gaussian distribution by the Central Limit Theorem. According to our experiment settings in this experiment, it takes 360 phases for a sensor to consume energy to the amount of the stair height (i.e., $\frac{e}{N_{max}}$). Since all area's coverage number complies to the same distribution, their summation of coverage numbers over a large number of phases can be approximated with the same Gaussian distribution with the same mean. Thus, all areas consume energy at approximately the same average rate.

Furthermore, our scheme maintains a phase difference δ among the staircases in different areas. Thus, the time instances for all areas to send out ready message are *approximately* evenly distributed as time evolves. Therefore, both average replacement interval and utility of the MR follow the similar pattern as in Fig. 5.

One notable difference between Fig. 6 and Fig. 5 is in the histograms of replacement intervals. The histograms in Fig. 5(c) are taller and narrower than the corresponding ones in Fig. 6(c), which implies smaller standard deviations. This is because the independence of coverage numbers of the areas brings more variance in terms of the interval between two consecutive time instances when the number of sensors that are ready to be replaced reaches x .

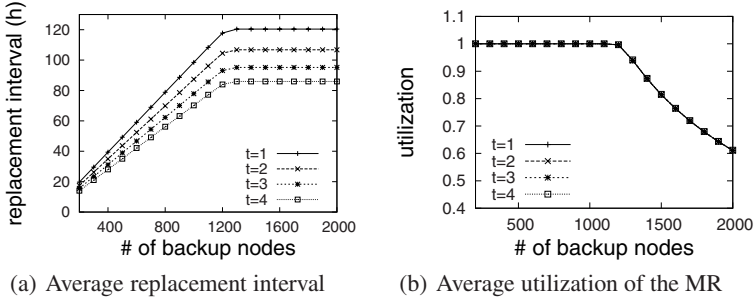


Fig. 7. Variable Gaussian distribution

6.4 Variable Distribution of Coverage Numbers

In this experiment, all areas have the same values of parameters N_{min} and N_{max} , and their coverage numbers comply to the same truncated Gaussian distribution and are independent of each other. In the prior experiments, we always truncate $Gau(\mu = N_{min}, \sigma = 2)$ to the range $[N_{min}, N_{max}]$ to get truncated Gaussian coverage numbers. In this experiment, we set $N_{min} = 1$, $N_{max} = 4$, and truncate $Gau(\mu = t, \sigma = 2)$ to the range $[N_{min}, N_{max}]$, where t varies in $\{N_{min}, N_{min} + 1, \dots, N_{max}\}$, i.e., $\{1, 2, 3, 4\}$. We only consider one backup coverage set in this experiment.

Fig. 7 shows the trend of average replacement interval and utilization of the MR when t varies.

As can be seen, when t is larger, average replacement interval is smaller. This is because larger t implies higher energy consumption rate of the network, and thus the MR needs to replace sensors more frequently. On the other hand, the value of x where average replacement interval levels off and the utilization of the MR starts to drop is the same for all the values of t . This is because the distribution of coverage numbers does not affect the upper bound of x according to Eq. (3).

7 Related Work

Recent studies [15, 16] have explored mobility to mitigate the energy issues. These schemes work in a “preventive” way and try to relieve sensor nodes from some responsibilities by leveraging mobile nodes. However, when a certain number of sensor nodes are drained of energy, the network cannot heal itself and thus cannot operate for a long time, which is required by long-term surveillance applications.

To enable self-healing of a sensor network and for other purposes, Wang et al. [17] introduce mobile sensors to replace sensors died of energy depletion. In a long-time surveillance application, eventually, all the sensor nodes need to be replaced by the mobile nodes, which increases the network cost. Schemes [18, 19] that propose to employ unmanned aerial vehicles or robots to repair networks have the following drawbacks: i) Infinite number of backup sensors is assumed. ii) Intensive communication between sensors and base station(s), and sensors and robots, is required.

Another approach to address the energy issues is to take advantage of ambient energy [3,5,6] in the environment, e.g., solar energy. As mentioned in Section 4, practical solutions are still under investigation.

8 Conclusion

In this paper, we proposed an on-demand node reclamation and replacement scheme for long-term surveillance sensor networks based on the area coverage model. Our scheme periodically replaces sensors drained of energy given a fixed number of backup sensor nodes, and guarantees that the coverage requirement of the network is satisfied over an infinite period of time. The simulation results show our scheme are both effective and efficient.

Acknowledgments

This work was partially supported by the National Science Foundation under Grands CNS-0831874, CNS-0831906, CNS-0834585, and CNS-0834593.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. *Computer Networks* 38(4) (2002)
2. Zeng, K., Ren, K., Lou, W., Moran, P.J.: Energy-aware geographic routing in lossy wireless sensor networks with environmental energy supply. In: *Proc. of QShine 2006*, Waterloo, Ontario, Canada (2006)
3. Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., Srivastava, M.: Design considerations for solar energy harvesting wireless embedded systems. In: *Proc. of IPSN 2005*, Los Angeles, CA, pp. 457–462 (2005)
4. Kansal, A., Hsu, J., Srivastava, M.B., Raghunathan, V.: Harvesting aware power management for sensor networks. In: *Proc. of DAC 2006*, San Francisco, CA, pp. 651–656 (2006)
5. Kansal, A., Srivastava, M.B.: An environmental energy harvesting framework for sensor networks. In: *Proc. of ISLPED 2003*, Seoul, Korea, pp. 481–486 (2003)
6. Kansal, A., Potter, D., Srivastava, M.B.: Performance aware tasking for environmentally powered sensor networks. In: *Proc. of ACM SIGMETRICS 2004*, New York, NY, pp. 223–234 (2004)
7. Tong, B., Wang, G., Zhang, W., Wang, C.: Node reclamation and replacement for long-lived sensor networks. In: *Proc. of IEEE SECON 2009*, Rome, Italy (2009)
8. Thai, M.T., Wang, F., Du, D.H., Jia, X.: Coverage problems in wireless sensor networks: designs and analysis. *International Journal of Sensor Networks* 3(3), 191–200 (2008)
9. Cardei, M., Thai, M.T., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: *Proc. of IEEE INFOCOM 2005*, Miami, FL, pp. 1976–1984 (2005)
10. Xing, G., Wang, X., Zhang, Y., Lu, C., Pless, R., Gill, C.D.: Integrated coverage and connectivity configuration for energy conservation in sensor networks. *TOSN* 1(1) (2005)
11. Zhang, H., Hou, J.C.: Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks: An International Journal* 1(1-2), 89–123 (2005)

12. Bai, X., Kuma, S., Xuan, D., Yun, Z., Lai, T.H.: Deploying wireless sensors to achieve both coverage and connectivity. In: Proc. of MobiHoc 2006, Florence, Italy, pp. 131–142 (2006)
13. He, T., Krishnamurthy, S., Stankovic, J.A., Abdelzaher, T.F., Luo, L., Stoleru, R., Yan, T., Gu, L., Hui, J., Krogh, B.H.: Energy-efficient surveillance system using wireless sensor networks. In: Proc. of MobiSys 2004, Boston, MA (2004)
14. Estrin, D., Govindan, R., Heidemann, J.S., Kumar, S.: Next century challenges: Scalable coordination in sensor networks. In: Proc. of MobiCom 1999, Seattle, WA, pp. 263–270 (1999)
15. Luo, J., Hubaux, J.-P.: Joint mobility and routing for lifetime elongation in wireless sensor networks. In: Proc. of IEEE INFOCOM 2005, Miami, FL, pp. 1735–1746 (2005)
16. Somasundara, A.A., Ramamoorthy, A., Srivastava, M.B.: Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In: Proc. of RTSS 2004, Lisbon, Portugal, pp. 296–305 (2004)
17. Wang, G., Cao, G., Porta, T.L., Zhang, W.: Sensor relocation in mobile sensor networks. In: Proc. of IEEE INFOCOM 2005, Miami, FL, pp. 2302–2312 (2005)
18. Mei, Y., Xian, C., Das, S., Hu, Y.C., Lu, Y.-H.: Sensor replacement using mobile robots. *Comput. Commun.* 30(13), 2615–2626 (2007)
19. Corke, P., Hrabar, S., Peterson, R., Rus, D., Saripalli, S., Sukhatme, G.: Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In: Proc. of ICRA 2004, New Orleans, LA, pp. 1143–1151 (2004)

Variable Density Deployment and Topology Control for the Solution of the Sink-Hole Problem

Novella Bartolini, Tiziana Calamoneri, Annalisa Massini, and Simone Silvestri

Department of Computer Science
“Sapienza” University of Rome, Italy
{bartolini, calamo, massini, simone.silvestri}@di.uniroma1.it

Abstract. The use of mobile sensors is of great relevance to monitor critical areas where sensors cannot be deployed manually. The presence of data collector sinks causes increased energy depletion in their proximity, due to the higher relay load under multi-hop communication schemes (sink-hole phenomenon). We propose a new approach towards the solution of this problem by means of an autonomous deployment algorithm that guarantees the adaptation of the sensor density to the sink proximity and enables their selective activation.

The proposed algorithm also permits a fault tolerant and self-healing deployment, and allows the realization of an integrated solution for deployment, dynamic relocation and selective sensor activation.

Performance comparisons between our proposal and previous approaches show how the former can efficiently reach a deployment at the desired variable density with moderate energy consumption under a wide range of operative settings.

1 Introduction

The deployment of mobile sensors is attractive in many scenarios. For example, mobile sensors may be used for environmental monitoring to track the dispersion of pollutants, gas plumes or fires. They may also be used for public safety, for example to monitor the release of harmful agents as a result of an accident. In such scenarios it is difficult to achieve an exact sensor placement through manual means. Instead, sensors may be deployed somewhat randomly from a distance, and then reposition themselves to provide the required sensing coverage. We formally prove the termination of our approach. The potential of such applications has inspired a great deal of work on algorithms for deploying mobile sensors. Most of this work has addressed the deployment of homogeneous sensors to achieve a uniform coverage of a certain density in a specific Area of Interest (AoI). When the sensor network centralizes the communications towards a single or a few sinks, the energy depletion due to communications is uneven and may possibly cause the so-called *sink-hole phenomenon* [1,2,3]. In this paper we address this practical and challenging problem by deploying sensors

at variable densities to ensure uniform energy depletion even under imbalanced communication load.

We propose an algorithm which is based on a generalization of the Push & Pull approach presented in [4]. In summary, our contributions are:

- We identify the models of load imbalance caused by centralized communications towards one or more sinks in the network and propose a density function that models the varying density requirements over the AoI as a consequence of those unbalanced communications;
- We propose a new algorithm based on the known Push & Pull algorithm so as to allow it a more direct control over the placement of redundant sensors, to provide a sensor deployment at variable controlled density;
- We extend a virtual forces based algorithm to operate in a scenario with variable density requirements, in order to make fair comparisons between our approach and the one based on virtual forces.

The Push & Pull algorithm is practical as it provides very stable sensor behavior, with fast and guaranteed termination and moderate energy consumption. It does not require manual tuning or perfect knowledge of the operating conditions, and works properly if the sensor positioning is imprecise. The algorithm does not require any synchronization during the deployment phase. The achieved deployment permits the use of alternate sensor activation that can be adopted if a loose synchronization is possible during the operative phase of the network. Because it converges quickly and does not require a priori knowledge of the deployment environment, it is also well suited for dynamic environments in which multiple sinks can be dynamically placed in consequence to dynamically changing missions.

The paper is organized as follows. Related work is presented in Section 2. In Section 3 we motivate the problem and introduce some preliminary concepts. Section 4 is the core of the paper and presents a new algorithm for variable density sensor deployment. In Section 5 we show how to exploit the described algorithm to jointly solve the problem of sensor deployment, dynamic relocation, self-healing and selective activation. Section 6 is devoted to summarize a virtual force based algorithm that we use to perform experimental comparisons whose results are shown in Section 7. Section 8 concludes the paper addressing some final remarks.

2 Related Work

Various solutions have been proposed to the problem of mobile sensor self-deployment. The majority of them are either based on the virtual force approach (VFA) or on computational geometry models. According to the VFA technique [5, 6, 7, 8] the interaction among sensors is modelled as a combination of attractive and repulsive forces. Other solutions [9, 10] have been inspired by different physical models. All these approaches require a laborious tuning of thresholds and constants to determine the magnitude of the forces and to

control possible oscillations. The choice of these values influences the resulting deployment, the overall energy consumption and the convergence rate.

Most of the deployment methods based on computational geometry model the deployment problem in terms of Voronoi diagrams or Delaunay triangulations [11,12]. Similarly to the VFA approach, these proposals rely on the off-line tuning of key parameters to avoid movement oscillations.

All the above mentioned solutions do not address the sink-hole problem. Only [13] presents a unified solution for sensor deployment and relocation crowding sensors in the presence of events. This approach could be adopted to increase the sensor density in proximity of the sink. On the contrary, papers dealing with the sink-hole problem explicitly, only focus on static sensor deployment [14,2,3,15].

The aim is to mitigate the effects of the uneven energy depletion due to communication with a sink by means of a variable density deployment. In the next section we will detail some of these results that will be useful for our contribution.

Many works deal with the k -coverage deployment problem. In [16], Vu and Verma reduce the problem of sensor placement with a redundancy of at least k sensors to the problem of distributing k points evenly on a torus manifold by minimizing the Riesz energy. In [17] the k -coverage sensor deployment problem is considered in both cases of the binary and probabilistic sensing models. They also distinguish the problem of sensor placement in the case of the different relation between the sensing radius r_s and communication radius r_c , i.e. $r_c < \sqrt{3}r_s$ and $r_c \geq \sqrt{3}r_s$ and propose two different dispatch schemes.

The k -coverage sensor placement can be obtained by shrinking a grid deployment until the k -coverage is achieved. In both [4] and [18] the shrinking is used to obtain a denser hexagonal grid.

In the present work a redundant coverage with adaptive redundancy level k , is obtained by superimposing several grid translated from each other to the purpose of achieving a variable controlled density deployment. Furthermore the k -coverage is exploited to the purpose of ensuring uniform energy depletion by performing a selective activation of the sensors.

3 Density Requirements in the Presence of Centralized Communications towards the Sink

Li and Mohapatra address the sink-hole problem in [2]. The authors analyze the applicative context of environmental monitoring and data gathering. In this context they assume that each sensor generates new traffic with a constant bit rate (CBR) and sends it to the sink via multi-hop communications. The examined deployment consists of a uniform random placement of devices over the AoI, where N is the total number of devices and A_{net} is the measure of the area of the AoI, hence the uniformly deployed density is $\rho = N/A_{net}$. Sensors transmit their packets to the destination by selecting the next-hop which is closest to the destination.

The authors propose a model to evaluate the per-node energy consumption, by considering three main contributions, namely energy spent for sensing, transmissions and receptions. They divide the AoI into several concentric circular crowns of radius equal to the transmission range r , centered at the sink position. The energy consumption of the sensors is then calculated separately in each crown.

According to this model the per-node energy consumption of the i -th crown is the following:

$$ECR_{ring^{ith}} = \alpha_1 b + \gamma_1 \frac{(\frac{M^2}{\pi} - (i+1)^2)}{2i+1} b + (\beta_1 + \beta_2 r^n) \frac{(\frac{M^2}{3} - i^2)}{2i+1} b \quad (1)$$

where $i = 0, 1, \dots, (\frac{M}{2} - 1)$, and the parameters are the following: b is the constant bit rate generated by each sensor, α_1 , β_1 , β_2 and γ_1 are technology dependent constant factors that are considered in the definition of the three energy contributions mentioned above, and the AoI is divided into $\frac{M}{2}$ concentric circular crowns with a step size of r meters.

Also Olariu and Stojmenović deal with the sink-hole problem in [3]. The authors also consider a uniformly deployed sensor network, with devices transmitting the same number of reports towards the sink. The authors conclude that the energy consumption of sensors located inside the i -th circular crown centered at the sink, and determined by the radii r_{i-1} and r_i , is as follows:

$$E_i = \frac{T}{\rho\pi} \left[1 - \frac{r_{i-1}^2}{r_i^2} \right] \frac{(r_i - r_{i-1})^\alpha + c}{r_i^2 - r_{i-1}^2} \quad (2)$$

where T is the number of tasks handled by the network during its lifetime, c is a technology dependent positive constant, $\alpha > 2$ is the power attenuation and ρ is the sensor uniform density over the AoI.

Finally the problem of uneven energy depletion due to many-to-one communications is addressed in [1] under nonuniform sensor deployment. The authors find a suboptimal deployment technique to ensure energy efficiency and mitigate the sink-hole problem. They propose to deploy sensors into circular crowns at different densities where the ratio between the sensor densities of the adjacent $(i+1)$ -th and the i -th crowns is equal to

$$\frac{\rho_{i+1}}{\rho_i} = \frac{(2i-1)}{q(2i+1)} \quad (3)$$

and $q > 1$ is the geometric proportion defining the increase in the number of sensors from the outer to the inner crowns. The circular crowns are centered at the sink position, and are dimensioned so as to ensure that the sensors of each crown act as forwarders for the outer crowns.

The authors assume a constant bit rate generated by each sensor and two energy contributions due to transmissions and receptions.

In this paper we refer to the above mentioned work [1] to define the non-uniform density requirements to be addressed by the deployment algorithm in

order to balance the energy consumption among the sensors of the network. By deploying the sensors according to Equation (3) the proposed approach ensures the network energy efficiency and prolong the network lifetime avoiding the generation of sink holes due to communications.

4 Variable Density Self Deployment of Mobile Sensors

The proposed algorithm, called δ -Push&Pull, is inspired by the algorithm introduced in [4], to which we made major modifications to the purpose of deploying sensors at variable densities according to position dependent requirements.

Given a point P in the AoI, we define $\delta(P)$ the coverage density required in position P . Let V be a set of equally equipped sensors able to determine their own location, endowed with boolean sensing capabilities and isotropic sensing and communication model. Notice that location capabilities are only necessary to recognize the borders of the AoI while, in order to make movement decisions, each sensor only needs to know the position of its communicating neighbors.

As in its original counterpart, according to δ -Push&Pull, the sensors aim at realizing a complete coverage of the AoI and a connected network by means of a hexagonal tiling deployment, where the side of each hexagon is set to the sensing radius r_s . The hexagonal tiling is realized by snapping the necessary number of sensors over the AoI in grid positions located in correspondence to the vertices of a triangular lattice with side $\sqrt{3}r_s$. Such sensors will be referred to as *snapped*. Given a snapped sensor x , we refer to $Hex(x)$ as to the hexagonal area that is covered by the sensor x and to P_x as to the position of the sensor x .

At the same time, δ -Push&Pull deploys redundant sensors over the covered area, by distributing them at variable density, according to $\delta(P)$ as follows: the number of sensors that will be located in $Hex(x)$ centered at P_x is $n_\delta(P_x) \triangleq \lceil \delta(P) \cdot \frac{3\sqrt{3}}{2} r_s^2 \rceil$.

The $n_\delta(P_x) - 1$ sensors utilized to obtain the desired density in a specific hexagon will be indicated as *adjunct-snapped sensors*. The sensors located in $Hex(x)$ which are neither snapped nor adjunct-snapped will be named *slaves* of x . We hereafter refer to $S(x)$ as the set of slave sensors of x .

The algorithm starts with the concurrent creation of several tiling portions. Every sensor not yet involved in the creation of a tiling portion gives start to its own portion in an instant which is randomly selected in a given time interval. Such a starter sensor is called s_{init} . The algorithm consists of four main interleaved activities: *snap*, *push*, *pull* and *merge*.

Snap Activity

The sensor s_{init} elects its position P_{init} as the center of the first hexagon of its tiling portion. It collects information on the sensors in radio proximity, that will compose the set $L(s_{init})$. Among the sensors located in its own hexagon, s_{init} chooses up to $n_\delta(P_{init}) - 1$ sensors for the role of adjunct-snapped. Such sensors will remain in their original hexagon and will not participate in the following activities. The sensors belonging to $L(s_{init})$ which have not been declared

adjunct-snapped can be used to cover adjacent hexagons. To this purpose, s_{init} selects at most six sensors among those belonging to $L(s_{\text{init}})$ and makes them snap to the center of adjacent hexagons. Such deployed sensors, in turn, give start to their own selection and snap activity, thus expanding the boundary of the current tiling portion. This process continues until no other snaps are possible, because either the whole AoI is covered, or the boundary tiles do not contain any unsnapped sensors.

Sensor x starts the push activity if slave sensors are still present in $Hex(x)$ after the adjunct-snapped declaration and the adjacent positions are all covered by snapped sensors. By contrast, sensor x starts the pull activity if (1) the number of adjunct-snapped sensors is lower than necessary to fulfill the density requirement, or (2) some hexagons adjacent to $Hex(x)$ are left uncovered and x has no slaves.

All the snapped sensors position the adjunct-snapped sensors in their hexagon according to a same common rule. This way it is possible to obtain the desired distribution of sensors over the hexagon area. Moreover, it is possible to perform a selective sensor activation which allows energy saving during the operative phase of the network, giving rise to alternate activation of different hexagonal grids composed by adjunct-snapped sensors in the same position. Obviously, these adjunct grids have the same coverage and connectivity features of the main hexagonal grid, that is the grid composed by the snapped sensors.

Push Activity

After the completion of their snapping activity, snapped sensors may have slave sensors located inside their hexagon. In this case, they pro-actively push such slave sensors towards the areas demanding a higher number of sensors. Consequently, slave sensors being in overcrowded areas migrate to zones with unsatisfied density requirements.

In order to avoid endless cyclic movements of slaves, we introduce the following δ -Moving Condition. The offer of slave sensors by a sensor x to a sensor y located in radio proximity is allowed if and only if:

$$\{|S(x)| > (|S(y)| + 1)\} \vee \{|S(x)| = (|S(y)| + 1) \wedge id(x) > id(y)\}$$

where $id(\cdot)$ is a function initially set to the unique identity code of the sensor radio device.

If the δ -Moving Condition is verified, sensor x can push at least one of its slaves towards the destination hexagon $Hex(y)$ selected as the one that needs a higher number of sensors to fulfill the local density requirements or to fill an adjacent coverage hole; among the slave sensors which can be pushed to the destination, x selects the closest to $Hex(y)$.

Pull Activity

The sole snap and push activities are not sufficient to ensure the maximum expansion of the tiling and the achievement of a deployment at the required density. In the δ -Push&Pull algorithm, the pull activity starts whenever a sensor

x notices either a hole in its adjacent snapping position or a density in $Hex(x)$ that is lower than $n_\delta(P_x)$.

Snapped sensors may detect a coverage hole adjacent to their hexagon and may not have available sensors to make them snap. Similarly, a snapped sensor may need more adjunct-snapped sensors than available to fulfill the density requirements. In these cases, they send hole trigger messages, and re-actively attract non-snapped sensors and make them fill the hole or the density gap.

In order to start the pull activity, sensor x broadcasts an invitation message at a higher and higher number of hops, until it receives an acceptance of invitation from a snapped sensor having a redundant slave. The inviter acknowledges the acceptance message if it has not found a number of slave sensors sufficient to fill the hole or the density gap, or reject it otherwise. In the former case, an agreement has been reached between the two sensors and the slave can start moving. When the snapped sensor that is performing the pull activity reaches its objective (to fill either the hole or the density gap), it stops sending slave invitation messages.

Merge Activity

The possibility that many sensors act as starters can give rise to several tiling portions with different orientations. In order to characterize and distinguish each tiling portion, the time-stamp of each starter is included in the header of all exchanged messages. Then, messages coming from sensors located in different tiling portions include different starter time-stamps. When the boundaries of two tiling portions come in radio proximity with each other, the one with older starter time-stamp absorbs the other one by making its snapped sensors move into more appropriate snapping positions. Hence this activity provides a mechanism to merge all the tiling portions into a unique regular and uniformly oriented tiling.

We conclude this description of the algorithm with an activity called **role exchange**. According to the previous description of δ -Push&Pull, slaves consume more energy than snapped and adjunct-snapped sensors, because they are involved in a larger number of message exchanges and movements. We introduce a *mechanism to balance the energy consumption* over the set of available sensors making them exchange their roles. This mechanism is similar to the technique of *cascaded movements* introduced in [19]. Namely, any time a slave has to make a movement across a hexagon as a consequence of either push or pull activities, it evaluates the opportunity to substitute itself with the snapped and adjunct-snapped sensors of the hexagon it is traversing. The criterion at the basis of this mechanism is that two sensors exchange their role whenever the energy imbalance is reduced. As a result, the energy balance is significantly enhanced, though the role exchange has a small cost for both the slave and the snapped sensor involved in the substitution. Indeed, the slave sensor has to reach the center of the current hexagon and perform a *profile packet* exchange with the snapped sensor that has to move towards the destination of the slave. A profile packet contains the key information needed by a sensor to perform its new role after a substitution.

4.1 An Example of the Algorithm Execution

Figure 1 illustrates the interleaved execution of the algorithm actions through an example. For simplicity, we do not consider the role exchange activity.

Figure (a) shows a starting configuration in which a sink is positioned in the central point of the right vertical side of the AoI and requires a density variation in its proximity. The sensor 8 assumes the starter role.

This sensor snaps three of its slaves, as shown in figure (b), where the *id* values of such snapped sensors are highlighted.

Figure (c) shows that the snapped sensor 8 has some un-snapped sensors in its hexagon, and therefore starts the push activity towards its three adjacent hexagons. In the meantime, the sensor 4 acts as starter and another grid portion is initiated. As it is in a zone with density requirement 4, it designates the sensors 20, 36 and 11 as adjunct-snapped.

In Figure (d) the snapped sensor 19 detects a coverage hole. As it has an un-snapped sensor in its hexagon, it performs the snap activity. The sensor 6 must satisfy a density requirement 2, so it designates the sensor 34 as adjunct-snapped. Notice that the snapped sensor 1 does not have any hole around its hexagon, so its slave remains where it is; furthermore, it does not execute any push action as the Moving Condition is not satisfied. The snapped sensor 8, having many slaves, continues its push activity. At the same time, the snapped sensor 4 snaps three of its slaves. Figure (e) shows that, while the snapped sensors 4 and 8 continue their push activities, the sensors 3 and 7 start the pull activity, as both detect a coverage hole and do not have any slaves to snap, so new sensors are snapped in the left grid.

In view of the pull activity, some sensors arrive in the hexagons of sensors 3 and 7, and become adjunct-snapped. The same happens in the right grid, with sensors 15, 28 and 31 – see Figure (f). The sensors 4 and 8 continue their push activity.

In Figure (g) the snapped sensors 4 and 8 continue their push activity while some new sensors are snapped. In the meantime, the snapped sensors in the zone with density requirement 4 designate some adjunct-snapped sensors.

As soon as the grid portions come in radio proximity with each other, the tiling merge activity is started (Figure (h)) and a unique grid is built. The adjunct-snapped sensors located inside the hexagon of the sensor 31 will change their status from adjunct-snapped to slaves, because the sensor 31 has been snapped outside the AoI in consequence of the merge activity. Finally, Figure (j) concludes this example, showing the last activities performed to completely cover the AoI.

5 Joint Solution to Sensor Deployment, Selective Activation, Self-healing and Dynamic Relocation

5.1 Selective Activation

Our approach relies on the availability of a sufficient number of sensors to cover each hexagonal tile at the required density, namely with a given number of

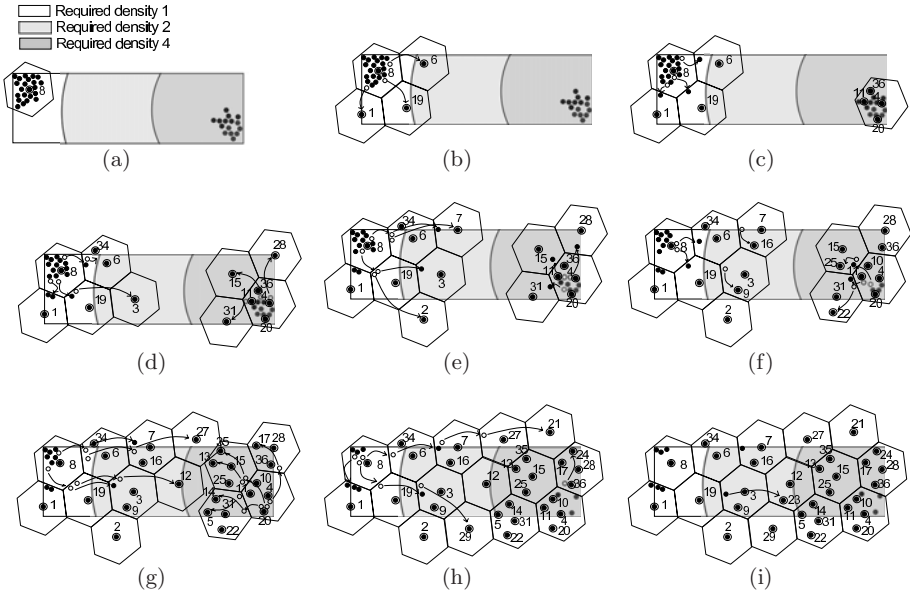


Fig. 1. Algorithm execution: an example

adjunct-snapped sensors. If the necessary number of sensors is available, the algorithm achieves a complete coverage, with a regular pattern that permits the use of topology control algorithms [20] and allows a selective sensor activation which saves energy during the operative phase of the network. As already highlighted, each snapped sensor will place its adjunct-snapped in fixed positions according to a predefined oriented pattern inside each hexagonal tile.

The deployment of the adjunct-snapped sensors according to the same pattern in each tile with the same density requirements, allows us to define a *selective activation pattern*. The selective activation of the sensors in a pattern guarantees the continuity and completeness of the coverage of the tiles that belong to the same circular crown.

When in an AoI there are crowns with different density requirements, temporary holes can appear along the boundary of these zones since sensors in different positions of the hexagons are activated in neighboring areas. This situation is described in Figure 2. Observe that the coverage discontinuity of Figure 2(b) is only intermittent, and many real applications may not suffer from it. Indeed, for some applications a continuous sensing of the AoI is not required, for example in the case of monitoring systems for the detection of pollutant levels, temperature or humidity conditions. In these cases, the monitoring activity can rely on the sole interpolation of local measurements taken at discrete points in the AoI.

By contrast, other more critical applications require that every point in the target area be accurately monitored, for example when the sensors are deployed to monitor the presence of human-life threats such as radioactive or chemical

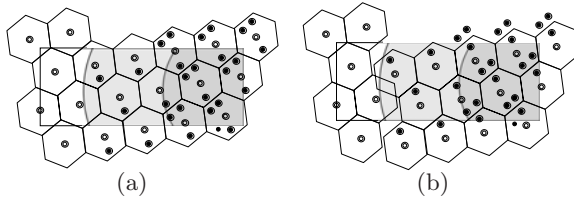


Fig. 2. Coverage holes at the borders of the circular crowns during the execution of the alternate activation of the adjunct-snapped sensors

plumes or a forest fire. In these cases, coverage discontinuities can be eliminated by positioning the adjunct-snapped sensors in the *wiggle region* of the snapped sensor. Indeed, the wiggle region has been defined in [18] as the region comprising all those points in which a sensor could be repositioned such that full coverage is maintained. Of course, the adoption of the wiggle region requires a slight shrinking of the hexagonal lattice. In particular, if w is the radius of the circle inscribed in the wiggle region, then the grid size must be set to $\sqrt{3}(r_s - w)$, instead of $\sqrt{3}r_s$. It follows that in order to create a wiggle region that is sufficiently large to accommodate all the adjunct-snapped sensors, it is necessary to deploy a larger number of sensors.

Notice that only a loose clock synchronization is actually necessary to perform the described selective activation scheme.

5.2 Self-healing and Dynamic Relocation

The proposed algorithm ensures that, when a sufficient number of sensors are available, the density requirements defined in correspondence to the center of each tile, will be fulfilled. Nevertheless, the algorithm does not give any indication on where to place redundant slave sensors, which instead are uniformly spread over the network as a consequence of the push activity. The redundant slave sensors will thus be available to recover possible failures. More in detail, as soon as a coverage hole is detected by the sensors located in proximity (for example, the detection may happen thanks to a periodic polling scheme or signalled by a failing sensor whose battery is almost exhausted), the detecting sensors can restart the algorithm with the consequence that the hole is immediately covered or a pull activity is executed to attract the closest slave sensors. The redundant slave sensors can thus be dynamically relocated to respond to pull invitations issued by the sensors located nearby failed devices. This process endows the network with self-healing and self-adapting capabilities that are not present in previous solutions.

In addition, a sensor network application may require sensor relocation capabilities (see [13, 19]) also to respond to dynamically occurring events when the deployment of new sensors is not possible, and the only choice is to re-use and move the available ones. In consequence of a dynamically occurred event, each

snapped sensor may declare a new density requirement, which better reflects the required position dependent accuracy.

This way the new set of redundant slave sensors become available to respond to new pull invitations necessary to reactivate the algorithm execution and fulfill the new density requirements.

6 On the Use of the Virtual Force Approach for the Deployment over a AoI with Variable Density

In order to evaluate the performance of the δ -Push&Pull algorithm proposed in this paper, we compare it with an algorithm based on virtual forces called Parallel and Distributed Network Dynamics (PDND), proposed in [21]. In PDND the force exerted by the sensor s_i on the sensor s_j is modelled as a piecewise linear function. It is repulsive when the distance between s_i and s_j is lower than an arbitrarily tuned parameter r^* ; it is attractive when the distance is larger, until it vanishes at another arbitrarily set distance. In order to ensure the convergence of PDND, the formulation of this force must respect the condition of Lipschitz continuity. In this case, the single sensor movement is limited by an upper bound that guarantees that the potential energy is always decreasing, hence avoiding oscillations.

PDND works under the assumption that density requirements are uniform over the AoI. In order to make the algorithm achieve a variable density deployment, we need to redefine the force that one sensor exerts on the others. According to the algorithm PDND, this implies the definition of the rest distance r^* at which the force exerted by two interacting sensors is null. More specifically, we assign to all sensors inside a region with the same density requirement a position dependent virtual sensing radius. In particular, we set the virtual sensing radius of a sensor as inversely proportional to the density requirement in its position. We consider a value of r^* that allows to minimize the overlaps among sensing disks, obtained as a combination of the sensing radii of two interacting sensors i and j , r_i and r_j , namely $r^* = r_i + r_j$. This value of r^* models the interaction between two sensors trying to position themselves so that their sensing circles are tangential.

It is to notice that the discontinuity of the density requirements over the AoI implies a discontinuity in the force function, that no longer respects the Lipschitz condition. For this reason, the convergence of the algorithm PDND is no longer guaranteed. In this particular setting, PDND loses its peculiar characteristic of guaranteed convergence and behaves as all the other algorithms based on virtual forces that, since the inspiring model is inherently dynamic, are prone to oscillations. In order to halt the execution of the PDND algorithm, we introduce a centralized oscillation control method as in [6]. By examining the history of movements of each sensor, we determine if oscillations are going on by checking if the sensor has moved back and forth around the same location many times. More formally, we say that a sensor is in an oscillatory state if in the last m movements it has not moved away more than ϵ_m meters from the barycenter of

such movements. We artificially terminate the algorithm as all the sensors are in an oscillatory state. We highlight that, although impractical, this oscillation control is of benefit for the performance of PDND and, for this reason, our comparisons are fair.

7 Simulation Results

In this section we compare our proposal with the PDND algorithm, adapted to our context as described in Section 6. To this purpose, we developed an OPNET based simulator. We use the following parameter setting: $r_{tx} = 10$ m, $r_s = 5$ m, sensor speed $v = 1$ m/sec. We consider a squared AoI of $120 \text{ m} \times 120 \text{ m}$ with three concentric circular crowns, centered at the sink position, located at the center of the AoI. According to [1], each crown has a different density requirement increasing geometrically towards the sink as described by Equation 3. In particular, we set the density requirement of the most external zone to one sensor per hexagon, and we use a parameter $q = 1.2$ for the geometric progression. In such a setting, the crown density requirements are 1, 2, 4 and 12 sensors per hexagon as we move from the outer to the inner crown.

We consider a random sensor initial deployment, as depicted in Figure 3(a). Figure 3(b) and 3(c) show an example of the final deployment achieved with 950 sensors by δ -Push&Pull and PDND, respectively. As it will be explained in the following, PDND achieves a more uniform deployment at the cost of a higher energy consumption and deployment time.

In order to compare the performance of the two algorithms we increase the number of deployed sensors from 800 to 1100. The results are obtained by averaging over 30 simulation runs.

Figure 4(a) shows the completion time, i.e. the time required to reach the final deployment. Recall that the PDND algorithm is artificially halted since it does not guarantee the termination. Despite this external intervention to halt the execution of PDND, the termination time of δ -Push&Pull is two orders of magnitude shorter than PDND. The slowness of PDND is due to the limitation to the distance each sensor is allowed to traverse at each round. On the other hand, δ -Push&Pull let sensors traverse entire hexagons at each movement, thus resulting in a shorter termination time.

Figure 4(b) shows the average traversed distance. δ -Push&Pull has a decreasing traversed distance as the number of sensors increases. This is due to the fact that less sensors have to be pulled in order to achieve the desired density as the number of deployed sensors increases. The PDND algorithm shows a higher traversed distance than δ -Push&Pull due to the oscillating movements typical of virtual force based solutions.

The average number of starting/stopping actions is shown in Figure 4(c). This is an important metric for mobile sensor deployment algorithms, because start and stop actions consume high energy [11]. PDND shows an average number of starting/stopping two orders of magnitude higher than δ -Push&Pull. As for the deployment time, this is due to the short distance each sensor can traverse

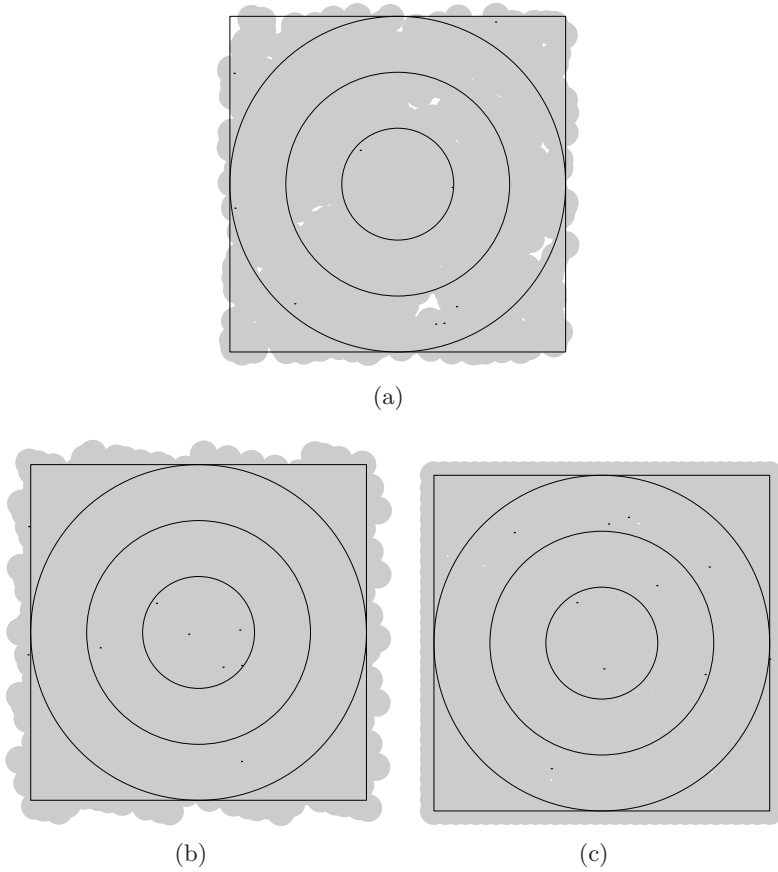


Fig. 3. Initial configuration (a). Final deployment under δ -Push&Pull (b) and PDND (c).

at each round. δ -Push&Pull, instead, moves the sensors precisely and without oscillations, resulting in a lower number of movements.

We now consider the average energy consumption of a sensor under the two algorithms. A sensor consumes energy due to communications (sending and receiving messages) and movements (travelling and starting/stopping movements). We consider two cumulative energy consumption metrics, namely the average energy spent in communication and the average total energy consumed by sensors. Such metrics are expressed in energy units: the reception of a message corresponds to one energy unit, a single transmission costs the same as 1.125 receptions [22], a 1 meter movement costs the same as 300 transmissions [11] and a starting/stopping action costs the same as 1 meter movement [11].

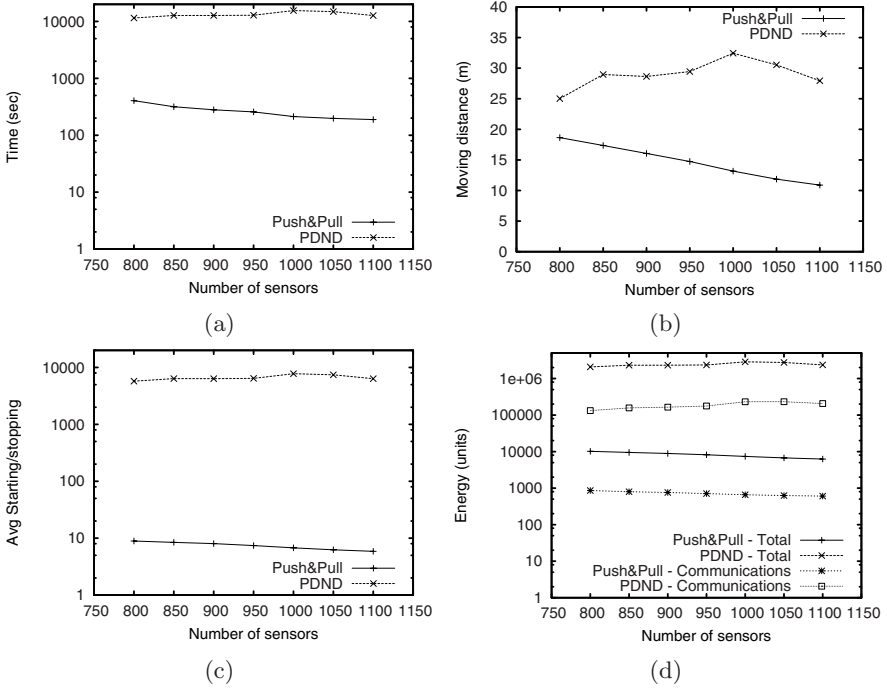


Fig. 4. Performance comparisons between δ -Push&Pull and PDND

Figure 4(c) shows the energy spent in communications and the total energy consumption. As expected, PDND has worse performance under both metrics. On the one hand, the energy spent in communications is higher because of the high number of rounds required by PDND to terminate. Indeed, under PDND, each sensor advertises its position to the neighborhood at each round. δ -Push&Pull, instead, has no round based communications, and messages are only exchanged to perform the algorithm activities. On the other hand, the higher number of starting/stopping actions as well as the higher traversed distance, result in a major total energy consumption of PDND with respect to δ -Push&Pull.

We finally evaluate the two algorithms considering the quality of the achieved deployments. We compared the percentage of AoI not meeting the desired density at the end of the algorithm execution. The results are shown in Figure 5. The regularity of the deployment achieved by PDND results in a better fulfillment of the requirements. However, such regularity is achieved at the cost of a higher energy consumption and a longer deployment time. δ -Push&Pull consumes two orders of magnitude less energy with respect to PDND, and is able to achieve a final stable deployment in a much shorter time. It shows a small gap in the percentage of area not meeting the desired density, that decreases as the number of sensors increases. This gap corresponds to the boundaries between adjacent circular crowns. Indeed, the density requirement of a tile is advertised according

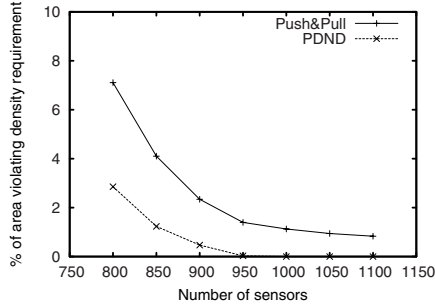


Fig. 5. Percentage of area not meeting the density requirements

to the position of its snapped sensor. Nevertheless, when a tile is crossed by the boundary line of a circular crown, one of the two sections lies on a crown where the density requirement is different from the one declared by the snapped sensor.

8 Conclusions

We proposed an original algorithm for mobile sensor self deployment, according to which sensors autonomously coordinate their movements to achieve a complete coverage with variable density. The sensor density varies so as to uniform the energy depletion due to communications towards the sink. The final deployment consists in a hexagonal tiling with a variable number of sensors deployed in each tile. Simulations show that our algorithm performs better than previous approaches in terms of several performance parameters. Furthermore we discussed some of the benefits related to the regularity of the obtained deployment. In particular we show how the regularity of the sensor distribution can be exploited to implement energy saving techniques and to achieve fault tolerance and self-healing capabilities.

References

1. Wu, X., chen, G., Das, S.K.: On the energy hole problem of nonuniform node distribution in wireless sensor networks. *IEEE Transactions on Parallel and Distributed System* 19, 710–720 (2008)
2. Li, J., Mohapatra, P.: Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. In: *Pervasive and Mobile Computing*, pp. 233–254 (2007)
3. Olariu, S., Stojmenovic, I.: Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting. In: *Proceedings of INFOCOM* (2006)
4. Bartolini, N., Calamoneri, T., Fusco, E., Massini, A., Silvestri, S.: Push & pull: autonomous deployment of mobile sensors for a complete coverage. *ACM/Springer Wireless Networks* (2009)

5. Zou, Y., Chakrabarty, K.: Sensor deployment and target localization based on virtual forces. In: Proc. IEEE INFOCOM (2003)
6. Heo, N., Varshney, P.: Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on Systems, Man and Cybernetics* 35 (2005)
7. Chen, J., Li, S., Sun, Y.: Novel deployment schemes for mobile sensor networks. *Sensors* 7 (2007)
8. Poduri, S., Sukhatme, G.S.: Constrained coverage for mobile sensor networks. In: Proc. of IEEE ICRA (2004)
9. Pac, M.R., Erkmén, A.M., Erkmén, I.: Scalable self-deployment of mobile sensor networks; a fluid dynamics approach. In: Proc. of IEEE IROS (2006)
10. Kerr, W., Spears, D., Spears, W., Thayer, D.: Two formal fluid models for multi-agent sweeping and obstacle avoidance. In: Proc. of the Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS (2004)
11. Wang, G., Cao, G., Porta, T.L.: Movement-assisted sensor deployment. *IEEE Transaction on Mobile Computing* 6 (2006)
12. Ma, M., Yang, Y.: Adaptive triangular deployment algorithm for unattended mobile sensor networks. *IEEE Transactions on Computers* 56 (2007)
13. Garetto, M., Gribaudo, M., Chiasserini, C.F., Leonardi, E.: A distributed sensor relocation scheme for environmental control. In: The ACM/IEEE Proc. of MASS (2007)
14. Wu, X., Chen, G., Das, S.K.: On the energy hole problem of nonuniform node distribution in wireless sensor networks. In: Proc. of IEEE MASS, pp. 180–187 (2006)
15. Cardei, M., Yang, Y., Wu, J.: Non-uniform sensor deployment in mobile wireless sensor networks. In: Proc. of WoWMoM, pp. 1–8 (2008)
16. Wu, C., Verma, D.: A sensor placement algorithm for redundant covering based on riesz energy minimization. In: Proc. ISCAS (2007)
17. Wang, Y.C., Tseng, Y.C.: Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. *IEEE Transactions on Parallel and Distributed System* 19 (2008)
18. Johnson, M., Sarioz, D., Bar-Noy, A., Brown, T., Verma, D., Wu, C.: More is more: the benefits of denser sensor deployment. In: Proc. INFOCOM (2009)
19. Wang, G., Cao, G., Porta, T.L., Zhang, W.: Sensor relocation in mobile sensor networks. In: Proc. of IEEE INFOCOM (2005)
20. Patten, S., Poduri, S., Krishnamachari, B.: Energy-quality tradeoffs for target tracking in wireless sensor networks. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 32–46. Springer, Heidelberg (2003)
21. Ma, K., Zhang, Y., Trappe, W.: Managing the mobility of a mobile sensor network using network dynamics. *IEEE Transaction on Parallel and Distributed Systems* 19, 106–120 (2008)
22. Anastasi, G., Conti, M., Falchi, A., Gregori, E., Passarella, A.: Performance measurements of mote sensor networks. In: Proc. of ACM MSWiM 2004 (2004)


QShine 2009

Session IV – Wireless, Mobility, and Context-Aware Services

iDSRT: Integrated Dynamic Soft Real-Time Architecture for Critical Infrastructure Data Delivery over WLAN

Hoang Nguyen, Raoul Rivas, and Klara Nahrstedt

University of Illinois at Urbana-Champaign, Urbana IL 61801, USA
{hnguyen5, trivas, klara}@illinois.edu

Abstract. The real-time control data delivery system of the Critical Infrastructure (i.e. SCADA - Supervisory Control and Data Acquisition system) is important because appropriate decisions cannot be made without having data delivered in a timely manner. Because these applications use multiple heterogeneous resources such as CPU, network bandwidth and storage, they call for an integrated and coordinated real-time scheduling across multiple resources to meet end-to-end deadlines. We present a design and implementation of *iDSRT* - an integrated dynamic soft real-time system to provide fine-grained end-to-end delay guarantees over WLAN. *iDSRT* takes the deadline partitioning approach: end-to-end deadlines are partitioned into multiple sub-deadlines for CPU scheduling and network scheduling. It integrates three important schedulers: *task scheduler*, *packet scheduler* and *node scheduler* to achieve global coordination. We validate *iDSRT* in Linux and evaluate it in an experimental SCADA test-bed. The results are promising and show that *iDSRT* can successfully achieve soft real-time guarantees in SCADA system with very low packet loss rate compared to available commodity best-effort systems. 

Keywords: Multi-resource scheduling, Quality-of-service, WLAN.

1 Introduction

Distributed real-time embedded (DRE) systems are key components of applications for critical infrastructures such as electric grid monitoring and control. These applications may use multiple heterogeneous resources such as CPU, network bandwidth and storage. For example, a Phasor Measurement Devices (PMU) in a power substation samples voltage and current at the rate of 60Hz. Sampled data is compressed and encrypted by an embedded processor and sent over a wired/wireless LAN. End-to-end delay of PMU data has to be guaranteed for real-time monitoring purpose. Another example is surveillance cameras

¹ This material is based upon work supported by the National Science Foundation under Grant CNS-0524695 and Vietnam Education Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of those agencies.

in a power substation (similar to [21]). A group of wireless cameras and sensors are placed around the substation for surveillance purpose. Each camera/sensor periodically captures a video frame compressed by an embedded processor and transmitted over a wired/wireless LAN. As shown in these examples, the heterogeneity and interactions of multiple resources in these applications call for an integrated and coordinated real-time scheduling across multiple resources to meet end-to-end deadlines. Unfortunately, even though scheduling for any single resource has been studied extensively, there has been little work done for integrated and coordinated real-time scheduling to meet end-to-end timing constraints (cf. see Section 6).

In this paper, we address the problem of integrated and coordinated scheduling of CPU and WLAN to meet end-to-end delay requirement. We use SCADA (Supervisory Control and Data Acquisition) systems for power substation monitoring as our case study. The general model of SCADA data WLAN is shown in Figure 1. This is a typical scenario specified in [1][2][3]. The scenario includes both real-time monitoring/control and non real-time management applications. Intelligent Electronic Devices (IEDs) periodically send sampling measurements (such as voltage, current, temperature) or video frames (for surveillance purpose) to a *gateway*. The gateway collects and processes sampling measurements (e.g. decompress, decrypt), issues necessary control actions to IEDs and reports necessary information to the control center. The delay requirement in this scenario is in the order of milliseconds [4]. In addition to the *real-time monitor and control* functionality, both the gateway and IEDs need to handle other *management* tasks. For example, the gateway may upload a configuration file to IEDs via a secure protocol (e.g. SSL).

We present a design and implementation of iDSRT - an integrated dynamic soft real-time system to provide fine-grained end-to-end delay guarantees over single-hop wireless networks. To guarantee end-to-end deadlines, iDSRT takes deadline partitioning approach. Specifically, end-to-end deadlines are partitioned into multiple sub-deadlines for CPU scheduling and network scheduling. The partitioning is done in such a way that the total system utilization is minimized for a given task set. To enforce sub-deadline guarantees at each scheduler, it employs

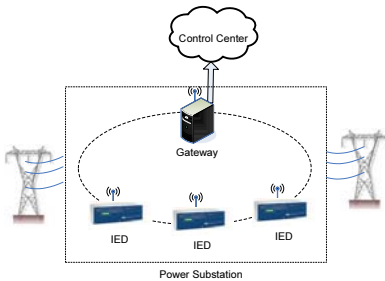


Fig. 1. SCADA data delivery deployment over Wireless LAN in a Power substation

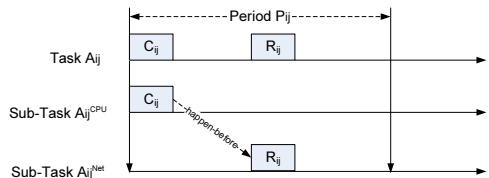


Fig. 2. Task Model

EDF (Earliest Deadline First) scheduling algorithm for both the *task scheduler*, called DSRT (Dynamic Soft real-time CPU scheduler), and the *packet scheduler*, called iEDF (Implicit EDF). The coordination between these two schedulers is executed by a *novel Coordinator entity*, called iCoord, sitting at the middle-ware layer. iCoord is the key component to deal with the inherent problem of scheduling for wireless network: the shared medium problem. Essentially, iCoord is a distributed node coordination scheduler that ensures every scheduler at each node coordinates with each other to meet end-to-end deadlines. Thus, iCoord plays the role of *node coordination scheduler*. Therefore, iDSRT has a unique approach: the integration of three important schedulers: *task scheduler*, *packet scheduler* and *node scheduler*.

In summary, our contributions in this paper are 1) the design of an *integrated architecture* with protocols and algorithms providing soft real-time end-to-end delay guarantees built on top of commodity Linux operating system and 802.11 MAC layer, 2) implementation of iDSRT including an augmented DSRT, iEDF and the Coordinator middleware and 3) performance study of iDSRT in an SCADA testbed of wireless nodes.

The rest of the paper is organized as follows. Section 2 presents our system model, notations and assumptions. In Section 3.1, we show the architecture of iDSRT and an overview of its components. Section 3.2, Section 3.3 and Section 3.4 give the details of iCoord, DSRT and iEDF. Section 4 presents necessary details of iDSRT implementation. In Section 5, we show our evaluation of iDSRT. Section 6 gives the related work and finally, Section 7 concludes the paper.

2 Models and Definitions

2.1 Network Model

We consider a single-hop wireless network model where each node is within one hop to the gateway as shown in Figure 1. There are n clients (i.e. IEDs) N_1, N_2, \dots, N_n and a server S (i.e. gateway). Client N_i has m_i ($m_i \geq 0$) real-time (RT) applications/streams and may have best-effort (BE) applications/streams running simultaneously. RT applications stream the data from the client to the server. Each RT application/stream will conform to its QoS specification in terms of end-to-end delay (EED) requirement.

EED is the sum of the delay at the sending side (i.e. at the client side), the propagation delay and the delay at the receiving side (i.e. at the server side). Controlling any of these components will affect EED. Our system, however, *only controls the delay at the sending side*. We assume the propagation delay is negligible compared to other two delay components. Furthermore, the receiving delay incurred at the gateway, including computation delay and MAC transmission delay, is small too. The reason is that we assume the gateway is a device with powerful computation and communication capabilities compared to the clients. Hence, controlling of this small delay component does not have much effect on the EED and it is also not the focus of our study.

The sending delay consists of the computation delay incurred by the OS scheduling and the communication delay incurred by the network scheduling. This delay component can be controlled by assigning deadlines to the computation and communication sub-tasks at each client (see Section 2.2). As long as these sub-tasks are finished on time by the OS scheduler and network scheduler, the EED requirements can be met.

In our model, BE applications may stream data to the server. These applications, if not monitored and enforced properly, can affect the QoS performance of other RT tasks because they are not aware of real-time constraints. Typical BE applications in a power substation are FTP application for downloading/uploading devices' configuration or data encryption for secure communication. These network- and computation-intensive applications may exhaustively consume network and CPU resources in the system if not constrained.

2.2 Task Model

We model the RT streaming applications as RT networked tasks, at the client side, composed of the computation and communication sub-tasks. The end-to-end delay requirement of streaming applications is now transformed into the *end-to-end deadlines* of the RT networked tasks used for scheduling.

Formally, we denote A_{ij} for the j th RT networked task/application on the client N_i where $i = 1..n$, $j = 1..m_i$. We also denote A_S as the networked task running on the server S . Each task A_{ij} has a period P_{ij} . It has two sub-tasks A_{ij}^{CPU} and A_{ij}^{Net} that needs to be processed in order (see Figure 2). That means, within period P_{ij} , the sub-task A_{ij}^{CPU} needs C_{ij} time unit for sampling and processing data. After the data gets processed, the sub-task A_{ij}^{Net} needs R_{ij} time units to send it to the server task A_S on server S over the wireless network G . The deadline D_{ij} of task A_{ij} is equal to the period P_{ij} . Both C_{ij} and R_{ij} are CPU and network resources consumed in time. C_{ij} is calculated by the number of consumed cycles over the CPU frequency. We assume the frequency of the CPU is fixed. Similarly, R_{ij} is the time of task A_{ij} and its underlying OS/network protocol stack to transmit a packet of size PS_{ij} bytes over the wireless MAC with measured bandwidth B_{ij} at node N_i , i.e. $R_{ij} = PS_{ij}/B_{ij}$ to the server S .

3 iDSRT Framework

3.1 Overview Design of iDSRT

Our first goal is to design/establish a scheduling and coordination framework of three important schedulers (i.e. the task scheduler, the packet scheduler and the node scheduler) that deliver end-to-end soft real-time guarantees in the system. The second goal is that the system should be able to run on a commodity platform (e.g. commodity Linux-based operating system and 802.11 MAC layer).

Each node N_i will consider time-sensitive scheduling of a) RT tasks A_{ij} , $i = 1..n, j = 1..m_i$ under competition of best-effort tasks, b) network packets of

² The terms "RT application A_{ij} " and "RT task A_{ij} " are used exchangeably.

connections belonging to the RT networked application A_{ij} and BE tasks at the node N_i and c) node N_i with respect to other nodes $N_k, k = 1..n, k \neq i$ due to the shared access to wireless medium.

The scheduling and coordination framework resides in the middleware, network and OS layers as shown in Figure 3 and it is called *iDSRT*. It allows RT and BE applications to run together and share resources in controlled manner. RT applications rely on iCoord (Integrated Coordination) - a distributed middleware component residing in the control plane of the protocol stack. It receives QoS specification from RT applications, performs RT application profiling, and does the QoS negotiation on behalf of the RT applications A_{ij} . Its central role is managing resource allocation within each node N_i and among nodes $N_i, i = 1..n$ and S in G to ensure end-to-end delay guarantees (see Section 3.2).

Any potential conflicts among RT tasks $A_{ij}, j = 1..m_i$ and BE tasks on node N_i are resolved by the Dynamic Soft-Real-time CPU Scheduler, called *DSRT* [15]. DSRT guarantees CPU resources for RT applications by using an adaptive EDF scheduling algorithm. It is “soft” because it does not manage other resources of the hardware and thus does not prevent the preemptions due to non-CPU hardware interrupts. However, the soft guarantees are within the timing bounds of SCADA tasks. Section 3.3 will give more details.

The last component in the iDSRT framework is the iEDF (Implicit Earlier Deadline First) packet scheduler. Essentially, iEDF is a network packet scheduler residing on-top of the MAC layer. It takes the implicit contention approach to schedule transmission slots according to the EDF policy. It manages the packet queue of each node and makes sure all nodes agree on the same packet to transmit over the shared medium within a specific time slot (see Section 3.4).

3.2 Integrated Middleware Coordination (iCoord)

iCoord is a distributed middleware component which coordinates all system scheduling components to ensure RT applications meet their deadlines. It operates in the control plane of the node’s protocol stack to provide the node registration service, task profiling and coordination services. Its services are a set of middleware libraries whose computation overhead is charged to the calling tasks’s computation. Figure 4 shows the middleware control architecture of iCoord. iCoord consists of two modules: *Local iCoord* residing on each client N_i and *Global iCoord* residing on the gateway S . Local iCoord is in charge of coordinating system components at each node N_i and communicates with Global iCoord to assist in inter-node scheduling with other nodes’ Local iCoord(s). Global iCoord executes global services on server S , where Local iCoord executes local services on each client N_i . Together, they ensure distributed utility services, such as the coordination service, registration service and the profiling service. Figure 7 summarizes the protocol within iCoord.

Registration Service is a service that takes care of the registration of real-time applications. Essentially, every RT application has to register with iDSRT because un-registered applications are treated as BE applications. First, the registration is done via the *Local iCoord Registrar*. The registration request

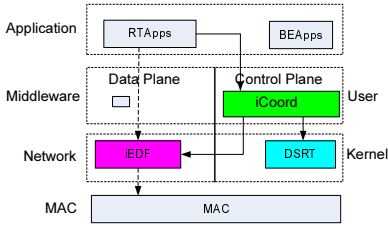


Fig. 3. End-to-End Integrated Dynamic Soft Real-time Framework (iDSRT)

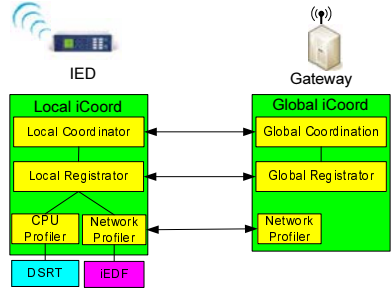


Fig. 4. Middleware control plane architecture iCoord

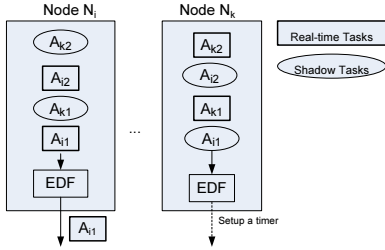


Fig. 5. Illustration of implicit contention scheduling

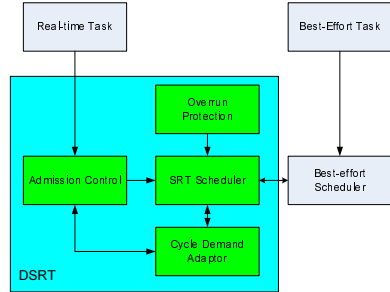


Fig. 6. DSRT Architecture

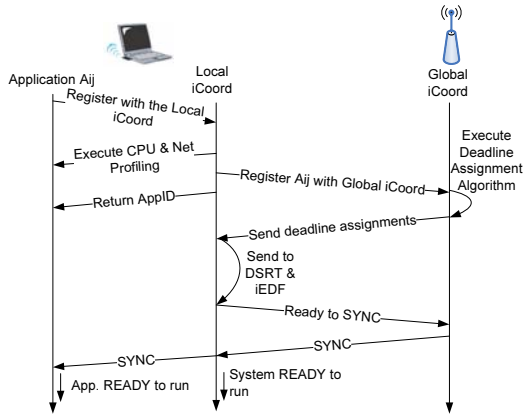


Fig. 7. iCoord protocol

from an RT application A_{ij} includes 1) a tuple of $(pid, saddr, sport, daddr, dport)$ where parameter pid , $saddr$, $sport$, $daddr$, $dport$ are the process identifier, the source address, the source port, the destination address, the destination port respectively. These parameters are used to uniquely identify each real-time communication application A_{ij} , 2) Period $P_{ij}(\mu s)$ and 3) a requirements $C_{ij}(\mu s)$ on CPU resource and network resource $R_{ij}(\mu s)$ measured by the profiling services.

The Local iCoord Registrar sends the registration information of this application to the *Global iCoord Registrar*. After the Global iCoord Registrar acknowledges the successful registration of the application A_{ij} , the Local iCoord Registrar returns a unique ID calculated from the tuple of registration information to the application. Finally, the Local iCoord Registrar invokes the CPU and network profiling services to approximate the CPU and network usage of the application (i.e. C_{ij} and R_{ij}). Finally, it sends the profiles of this task to the Global iCoord Registrar so that the node admission control, inter-node scheduling and coordination can be performed.

Profiling Service consists of the CPU and network profiler on each client N_i . These profilers are invoked after the registration phase. The CPU usage is measured by having DSRT run several instances of RT task A_{ij} . Similarly, the network profiling is done by measuring the packet round-trip-time between the networked application at the client N_i and the server S .

Coordination service is a distributed middleware component. Similar to the Registration service, the Coordination service has a Global Coordinator at the gateway S and a Local Coordinator at each node N_i . The Global Coordinator at the gateway gathers profiles of all RT applications from the Global Registrar and performs the deadline assignment algorithm (discussed in the next subsection). Then, it sends this information to all *Local iCoord Coordinators*. The information includes deadline assignments for the inter-node (i.e. D_{ij}^{Net}) and intra-node (i.e. D_{ij}^{CPU}) scheduling of all the tasks in the system G .

Upon receiving the deadline assignment of all tasks, the Local Coordinator confirms with DSRT and iEDF about the acceptance of these local tasks. At the end of this phase, each Local iCoord Coordinator notifies the Global iCoord Coordinator that node N_i is ready, and all local components DSRT, iEDF and Local iCoord wait for the SYNC message from the Global iCoord Coordinator. In the last phase, the Global Coordinator waits for all acknowledgments from Local Coordinators and broadcasts the SYNC message. The SYNC message start the run-time of the whole system.

Deadline Assignment Problem: As mentioned in the previous section, we employ the EDF algorithm for CPU scheduler (DSRT) and the network scheduler (iEDF). These two schedulers (DSRT and iEDF) must coordinate with each other so that the end-to-end deadline of RT applications A_{ij} can be met. The approach we take is partitioning the end-to-end deadline into sub-deadlines for the CPU scheduler and network scheduler. Thus, as long as the CPU scheduler and the network scheduler can schedule the sub-tasks correctly, the end-to-end deadlines will be guaranteed. The deadline assignment algorithm is executed by

the Global Coordinator whenever there is a newly arrival task. It is essentially a convex optimization algorithm where the deadline is split such that the total stress factor of the two sub-systems is minimized while still satisfying the admission control criteria of the CPU scheduler and the network scheduler. Please refer to our technical report for more information.

3.3 DSRT (Dynamic Soft Real-Time Scheduler)

DSRT is responsible for CPU task scheduling according to their deadlines. Specifically, on client N_i , it manages real-time CPU tasks A_{ij}^{CPU} , $j = 1..m_i$ as modeled in Section 2.2. To achieve this objective, DSRT is composed of three basic components, the Admission Control, the Earliest-Deadline-First (EDF) Scheduler and the Cycle Demand Adaptor.

On a node N_i , before using the realtime capabilities of the system, a new RT task A_{ij}^{CPU} must register itself with iCoord as a RT task in the DSRT. Specifically, it must specify its period, its worst case execution time and its relative deadline³. The admission control for DSRT on a node N_i is the EDF schedulability test. It means, $\forall L \in DLset, L \geq \sum_{j=1}^{m_i} (\lfloor \frac{L - D_{ij}^{CPU}}{P_{ij}} \rfloor + 1) C_{ij}$ where $DLset = \{d_{kl} | d_{kl} = lP_{ik} + D_{ik}^{CPU}, 1 \leq k \leq m_i, l \geq 0\}$ is the set including all tasks' deadlines less then the hyper-period of all periods (i.e. least common multiplier of P_{i1}, \dots, P_{im_i}).

If the condition is met, the task A_{ij}^{CPU} is added to the running queue of the EDF Scheduler and is scheduled to run in the next period. If the task cannot complete its job in the allotted time, due to demand cycle variations, the *Overrun Timer* will preempt the task to best-effort mode. In this case, the task A_{ij}^{CPU} will only be allowed to run after all other real-time tasks have used their allotted CPU time. The Overrun Timer removes the task from the running queue and adds it to the overrun queue. Tasks in best-effort mode compete against each other and use the standard OS non-realtime scheduler (Linux in the case of our implementation). Therefore, they cannot get a guaranteed CPU allocation.

If the deadline D_{ij}^{CPU} is not met, the Cycle Demand Adaptor will keep track of this event. If it detects that the change in the cycle demand is persistent and that assigned deadlines are not met, it will try to increase the allotted cycle demand for this particular task A_{ij}^{CPU} . In that case the Cycle Demand Adaptor will query the DSRT admission control to verify whether there are enough CPU resource to increase the allotted resource for the task A_{ij}^{CPU} .

3.4 iEDF (Implicit Earliest Deadline First Packet Scheduler)

iEDF is a distributed network scheduler that takes an ‘‘implicit contention’’ approach to perform the EDF packet scheduling algorithm [7][8]. Each client uses iEDF as its network scheduler. Conceptually, this network scheduler is actually an *outgoing-packet scheduler* working on top of the MAC layer. It manages how packets are prioritized to ensure they will meet the deadlines. Technical information will be given in Section 4.2.

³ The information C_{ij} and D_{ij}^{CPU} is provided by iCoord as explained in Section 3.2.

iEDF is an implicit contention scheduling which uses EDF as the packet scheduling algorithm. At any time slot, all clients agree on a RT task A_{ij}^{Net} to access the shared wireless medium according to the EDF policy. Specifically, for a client N_i , RT tasks $A_{ij}^{Net}, j = 1..m_i$ running on N_i are called *local RT network applications* and other RT applications running on other clients are called *remote RT network applications*. iEDF at each client N_i maintains the deadline assignment and task information of remote RT network tasks in addition to its local RT network tasks disseminated via iCoord (see Section 3.2).

Once iEDF has all network task deadline information, it creates a “shadow network task” for each remote network task. The shadow network task has the same period, deadline and transmission time as the network task being shadowed. When the shadow network task $A_{kj}^{Net}, j = 1..m_k$ “executes” on $N_i, i \neq k$, it does nothing but sets up a timer to wake up after the transmission of A_{kj}^{Net} . On waking up, the shadow network task again notifies iEDF that the remote network task is supposed to finish. On this event, iEDF schedules another RT network task, either local or remote (shadow) for the next transmission. In this way, iEDF is doing the EDF scheduling algorithm in a distributed manner. Furthermore, packet collisions will rarely happen because iEDF at each client aims to ensure and comply to the global deadline assignment. Figure 5 shows an illustration of this implicit contention.

Even though the principle of iEDF is simple, there are couple of issues that we need to address. The first issue is the correct estimation of the transmission time of the shadow network task. For any particular transmission, the remote network task A_{kj}^{Net} may finish earlier than expected due to worst case profiling and estimation of R_{kj} . It may also finish later than expected due to the noisy and unreliable channel. In the former case, iEDF ignores the early transmission and accepts the waste of idle network resource. In the latter case, iEDF actually has to avoid starting another transmission to minimize the packet collisions. To resolve this issue, iEDF only needs to over-hear the wireless network to know when the remote network task finishes. This is a simple solution yet enough to resolve the scheduling issues. The second issue is that even though iEDF is a network scheduler and consumes non-negligible CPU resource for scheduling. To resolve this issue, we let the network task’s CPU consumption to be charged to the computation time of the corresponding RT applications.

4 Implementation

4.1 DSRT Implementation

DSRT was originally implemented by Chu et al. [15] in Linux Kernel 2.4. Due to incompatibilities with Linux Kernel 2.6, DSRT is implemented from scratch in Linux Kernel 2.6. However, our implementation of DSRT is considerably different and includes important contributions to the original work. The main contributions are discussed below.

DSRT originally used the Liu and Layland scheduling model [17], in which the deadlines are considered to be equal to the periods. The coordination algorithm in iDSRT requires a more generalized model in which *the real-time scheduler supports deadlines less than or equal to the periods*. Our implementation of DSRT uses this model instead. Other important difference is that our implementations used new mechanisms developed for precise task accounting, including the CPU timestamp counter available in most modern processors and the new High-Resolution Timer Interface available in the most recent versions of the Linux Kernel 2.6 [10]. The use of this new mechanism allowed us to reduce the number of modifications to the standard kernel. It also allowed us to provide better precision and scheduling granularity than the previous implementation.

DSRT implements nine new system calls allowing RT tasks to communicate with it. These system calls provide DSRT with information required to reserve CPU resource and prioritize a task according to its QoS requirements. In these system calls the task A_{ij}^{CPU} specifies average cycle demands used to calculate C_{ij} (dividing by the CPU frequency), deadline D_{ij}^{CPU} and period P_{ij} . DSRT provides information about the performance and the status of the RT task, including the number of times a task tried to overrun and the statistical CPU utilization.

Our DSRT implementation needs only one kernel patch on the file *sched.c* to provide CPU accounting for each task. Linux currently provides such mechanism in the kernel but only with maximum resolution of 1 *jiffy* (number of iterations of the kernel per second)⁴ while we need high precision task accounting to the microsecond resolution. Simply increasing the *jiffy* resolution will cause enormous kernel overhead. In our implementation, we measure CPU usage of real-time tasks in cycles instead of jiffy. This is achieved by adding a hook in *schedule()* function. This hook is called every time that a context switch is about to occur. It allows us to measure the elapsed cycles between the current and the previous context switch and therefore precisely account for the CPU time of each task. Once done, the number of cycles is converted to time unit by dividing the number of cycles by CPU frequency. The rest of the DSRT is implemented as a kernel module. We use high-resolution timers provided in the kernel to ensure that tasks wake up at precise time and to prevent overruns from greedy BE and RT tasks. The context-switching is implemented as a two-halves operations where interrupts from the timer signals a high-priority kernel thread to preempt the running application.

DSRT has a new data structure to store the QoS parameters $C_{ij}, D_{ij}^{CPU}, P_{ij}$ of the task containing information about the state of the RT task used by both the EDF scheduler and the Cycle Demand Adaptor. When a new RT task makes a request for QoS guarantess to DSRT, DSRT creates a new instance of this data structure (called *srt_task_struct*) containing information about the state of the particular RT task. This task is also cross-referenced with the *task_struct* structure defined by the Linux scheduler to ensure proper communication between the Linux scheduler and DSRT. More precisely, the data structure contains a pointer to the associated *task_struct* structure, necessary information about the

⁴ Within Linux 2.6.10, a jiffy is by default 4ms.

state of the RT task in the DSRT scheduler⁵, the period, the cycle demand requested, the number of deadlines missed, the number of periods in which the task tried to overrun and the statistical CPU usage in cycles.

Conceptually, DSRT implements 3 *runqueues* that allow the EDF scheduler and the overrun timer to schedule the tasks. The first runqueue is for the RT task process currently ready to run. The second one is for the RT task processes that are running in best effort mode because they overrun. The last one is for the RT task processes that are awaiting for the beginning of the next period. We implement these runqueues as a single list of processes sorted by the EDF policy. We use the information stored in the *srt_task_struct* about the state of the task to differentiate among different runqueues. We avoid implementing more runqueues due to unnecessary kernel overhead. The computational complexity of the DSRT scheduler is $O(\log(n))$, where n is the number of RT tasks.

To further minimize the number of changes required to the Linux scheduler, the DSRT scheduler does not load or schedule the RT tasks directly, instead it relies on the Linux scheduler. The DSRT simply rises the priority of the running RT tasks to the highest RT priority available on the system, and requests a reschedule to the Linux kernel. This triggers a context switch and forces the Linux scheduler to pick the task that DSRT wants to be scheduled next. To preempt a running RT task to best effort mode when the overrun timer expires, it simply suffices to lower RT task's priority in the Linux scheduler to normal and rise the priority of another RT task. When all the RT tasks have completed the running job, they yield the CPU by invoking the *sched()* function. Upon the call, the Linux scheduler will take care of scheduling all the BE tasks including iDSRT aware and non-iDSRT aware tasks. The DSRT scheduler remains idle until one of the RT tasks begins a new period. This approach makes the implementation simple and ensure maximal compatibility with non-iDSRT aware tasks.

4.2 iEDF Implementation

iEDF is a queuing discipline in Linux. It communicates with the Local Coordinator via the */proc/* file system. This interface includes *create/modify/delete* a local (shadow) task A_{ij}^{Net} and a *SYNC* signal with the Local Coordinator. iEDF maintains the information of the tasks A_{ij}^{Net} by a double linked list data structure. Each shadow task in *iEDF* is implemented as a kernel timer that simulates the same behavior as the corresponding task. Note that even though iEDF may have many timers for shadow tasks, Linux Kernel 2.6 implements these timers as a single high resolution kernel timer to reduce the overhead.

iEDF maintains a FIFO queue for packets of each application. Each entry in a FIFO queue is a pointer to *sk_buff* kernel data structure of a real packet. Thus, iEDF works on pointers to packets to avoid any extra data copy overhead. In addition, iEDF maintains a bitmap representing applications that have packets in the FIFO queues (i.e. one bit per application). Whenever a packet of a RT

⁵ Note that a RT task A_{ij}^{CPU} can also become BE task if it violates its assigned deadline D_{ij}^{CPU} (cf. Section 3.3).

application enters the queue, the corresponding bit is set to 1. This bit will be set to 0 when the last packet leaves the corresponding queue. To look up applications with non-empty FIFO queue, iEDF uses $_ffs()$ operator on the bitmap to efficiently search for the 1-bit. Similar to DSRT, the complexity of iEDF scheduling is $O(\log(n))$, with n equal to the number of RT tasks.

5 Evaluation

5.1 Experiment Setup

We evaluate iDSRT in a SCADA test-bed of 7 nodes. Each node is a IBM T60 Dual Core 1.66Ghz laptop with 802.11a/b/g Atheros-based wireless card. We disable one core to emphasize the impact of DSRT on CPU scheduling. All laptops run Linux kernel version 2.6.16 with high-resolution timer patch.

We setup the laptops as shown in Figure 8. There is one laptop acting as a gateway. The rest of the laptops are clients. These laptops are emulated IEDs in a wireless SCADA testbed. Each client laptop connects to a real IED such as a Digital Relay and a Phasor Measurement Unit via serial ports. The gateway laptop connects to a SCADA server via Ethernet. To ensure that the gateway is more powerful than the clients, we set the CPU frequency of the server to the highest one (1.66 Ghz) and the CPU frequency of the clients to 1Ghz. All of laptops operate on 802.11a mode and are placed such that they are within the transmission range of each other. The network operates on the channel that has least interference to minimize external effects.

5.2 Scenarios

The RT application in the experiment is a regular periodic task creating/reading and sending a packet every 30ms. This is a typical RT task and time requirement for IED devices measurements (e.g. Phasor Measurement Unit devices) as specified in [4]. Each packet encapsulating a PMU measurement with the RTP-like header has the size of 128 bytes. The RTP-like header contains the sequence number for calculating packet losses and time-stamp for clock synchronization and delay calculation. Other parameters of RT applications such as computation time, network transmission time, sub-deadlines are measured and assigned by iDSRT. To scale up the experiments, each client may have more than one RT application. Specifically, we keep adding the RT applications in the system until the admission control fails. Note that for any number of RT applications in the system, these RT applications are distributed equally to clients. For example, if there are 10 RT applications in the system, each client has $\lfloor 10/6 \rfloor = 1$ application and the remaining four are assigned to any four clients.

Our goal is to make sure that the system can provide real-time guarantees even there are competing BE applications. On each client we run a very CPU-intensive BE application to compete for the CPU resource. Furthermore, we setup three BE TCP flows from three clients to the server. These three TCP flows will always try to send as much as they can when getting a chance.

5.3 Evaluation Metrics

We compare our iDSRT system against three other systems. The first one, named as “BestEffort” is the combination of commodity Linux and 802.11 MAC. The second one, named as “DSRT only”, is the system with DSRT enabled and iEDF disabled (i.e. DSRT and 802.11 MAC) and the last one, named as “iEDF only” has iEDF enabled only and DSRT disabled (i.e. Linux CPU scheduler and iEDF). The metrics we use are 1) RT end-to-end delay from a client N_i to the gateway S , and 2) the percentage of packet losses of RT applications A_{ij} and 3) the percentage of missing deadlines of RT applications A_{ij} .

All measurements above are done at the gateway S . In every experiment, each application sends 1000 samples, which takes $30ms \times 1000 = 30s$ to finish. The end-to-end delay is measured as the time difference from the packet sent at the client to the time that packet received at the server⁶. The percentage of packet losses are measured by counting the missing sequence numbers. Similarly, the percentage of missing deadlines is measured as the number of packets that are received later than the deadline at the gateway. Also, in each scenario, experiments are repeated 5 times to get the average measurements.

5.4 Experiment Results

End-to-End Delay: Figure 9 shows the end-to-end delay of different systems. The x-axis represents the total number of applications. The y-axis shows the average end-to-end delay in *ms*. In general, only iDSRT can guarantee the deadlines while the other systems cannot. BE system cannot handle the RT applications well because it does not prevent CPU-intensive application and TCP flows from exhausting CPU and network resources. This makes sense because BE system is designed for general purpose, not for real-time purpose.

DSRT-only system prioritizes RT processes and schedules them according to their deadlines. The CPU resource for RT processes is “reserved”, i.e. BE processes cannot compete for that reserved resource. That is the reason why DSRT-only system performs better than the BE system. However, because DSRT-only system can only provide RT guarantee on the CPU resource and it lacks of the RT network scheduler, the end-to-end delay cannot be guaranteed.

iEDF-only system performs worst due to two reasons. The first one is the lack of support from the RT CPU scheduler (i.e. DSRT). The BE CPU scheduler (i.e. Linux scheduler) is done in a round-robin fashion and is not aware of application deadlines. Consequently, packets arrive to iEDF in an aperiodic fashion, which may be earlier or later than the slots iEDF reserves for network transmission. If a packet of a RT task arrives to iEDF later than the slot reserved for its network part, it can only be transmitted until the next reserved time slot releases. This causes cascading missing deadlines of a RT task and can only be fixed with a coordination from the CPU scheduler. This explains why it performs worse than the BE system. The other reason is the shared nature of wireless medium among

⁶ The clocks of these clients are synchronized on every wireless transmission.

clients which makes consumed network resource grow quickly as the number of RT applications increase. That is why iEDF does not scale as good as DSRT.

iDSRT with the integration of DSRT, iEDF and iCoord performs the best. This result validates our design idea in which node scheduler, packet scheduler and task scheduler have to coordinate very well. Missing any components will not be sufficient to provide soft real-time guarantees.

It is also important to emphasize that iDSRT needs a good profiling and an admission control. In our scenario, iDSRT cannot accept more than 13 RT applications due to the admission control. We did run more than 13 applications and the results basically show that, not admitted RT tasks perform much worse than in BE system because most resources in iDSRT are reserved for admitted RT tasks. This, again, underscores the importance of QoS guarantees: once RT tasks are accepted, they will receive what promised by the system.

Missing deadlines: Figure 10 shows the percentage of missing deadlines of the four systems under various total number of applications. The x-axis shows the total number of applications. The y-axis is the percentage of missing deadlines. Generally, BE system and DSRT-only system have similar percentages of missing deadlines (30% to 40%). In these systems, the main cause for missing deadline is the lack of network scheduling. iEDF-only performs worst as expected due to the lack of support from CPU scheduling and higher resource-consuming rate of each application. iDSRT performs best and has only around 15% missing deadlines.

This figure also shows the nature of “soft” RT guarantees. The reasons for missing deadlines, even in the case of iDSRT, are preemptions due to hardware interrupts, non-preemptive nature of network scheduling (i.e. iEDF cannot preempt a packet being sent) and the unreliable nature of the wireless medium. These are also the reasons for the small fluctuations in the graphs. However, iDSRT with a low average end-to-end delay (around 15ms) and a reasonable missing deadlines (around 15%) is still the best to achieve soft RT guarantees.

To further support the need for the coordination, we show the maximum delay of each systems in Table 1. This table essentially shows the worst end-to-end delay of each system in our experiments. It is clearly shown that iDSRT has the smallest worst end-to-end delay with the deviation of 5ms. In other words, iDSRT misses 15% of deadlines but the deviation is *bounded* in 5ms.

Packet Losses: Figure 11 shows the packet losses of four systems. All systems have very low packet loss rate (less than 0.1%). iEDF in this case shows the

Table 1. Maximum EED (ms)

#Apps	Best Effort	DSRT-only	iEDF-only	iDSRT
6	90.06	90.54	51.58	34.17
7	93.75	88.40	69.90	35.38
8	96.88	92.34	82.82	35.59
9	482.05	101.30	102.43	33.78
10	508.31	109.15	138.21	33.59
11	505.76	97.77	154.13	34.17
12	516.69	128.25	164.42	34.63
13	558.37	136.11	169.35	35.25

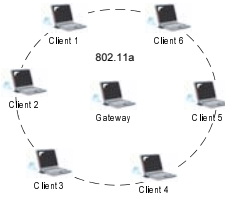


Fig. 8. Wireless SCADA testbed setup

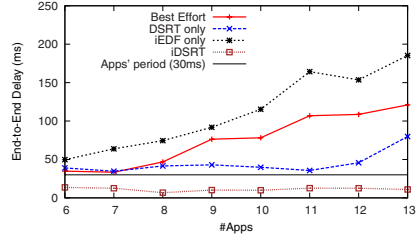


Fig. 9. End-to-End Delay

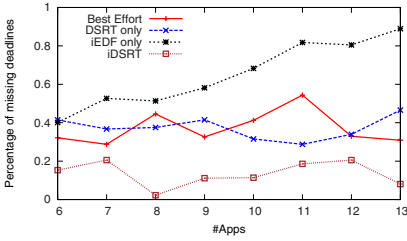


Fig. 10. Missing deadlines

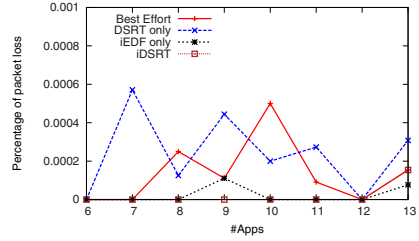


Fig. 11. Packet Loss

advantage of very low (almost no) packet losses due to the distributed scheduling mechanism. iDSRT, again, inherits the advantage of both DSRT and iEDF to achieve almost no packet loss.

6 Related Work

There has been large amount of research work that address individual, part of the end-to-end RT problem and can be classified into three categories: real-time operating system, real-time wireless network and end-to-end delay guarantee. The first category addressing real-time operating system has been extensively explored. Typical work in this category includes hard real-time solutions such as RTLinux [6]; firm real-time solutions such as Rialto [16], SMART [19], KURT [23] and soft real-time solutions such as DSRT [15], GraceOS [27]. RTLinux [6] is a period scheduler but it does not have admission control, the scheduler is a priority-based and non-preemptive. Thus, it does not support applications having deadlines and does not fit in our framework. The Rialto OS [16] focuses on the reservation of the resources. Its application model does not consider deadlines or periods but rather utilization constraints that the system must meet. The Resource Kernel is similar to our proposed work, they provide feedback to the application and use the concept of deadlines and periods. Their approach is not to coordinate the different reservations but to use a priority inheritance algorithm with a bounded waiting time. SMART [19] uses an EDF scheduler and a weighted

fair queue scheduler. It uses the Liu and Layland model [17], however they do not consider other resources or any coordination. KURT [23] uses a high-resolution timing system and a tickless kernel. However, it does not provide any guarantees and does not use admission control or reservation of resources. GraceOS [27] is a power-aware soft real-time OS. Its goal is to minimize the power usage of the system based on the QoS constraints specified by each application.

The second category addressing real-time wireless network has been also well explored. A typical work is the 802.11e standard [5]. Although 802.11e becomes the standard and commercially available, its prioritization mechanism does not work well when there are multiple flows with the same priority. In fact, it even increases more collisions due to its aggressive medium access parameters such as smaller CW_{min} and CW_{max} . Besides 802.11e, there are plethora of work involving with MAC design such as dynamic contention adaptation [25] [26], RI-EDF [8] or Wireless Token Ring [9]. Even though these schemes can work well, they require MAC modifications. iDSRT requires no modifications of the MAC layer and thus has the advantage of deployability and compatibility. We believe that with the wide-spread availability of 802.11-based hardware, it is much cheaper and more applicable to have a solution working on top of and independent of 802.11 MAC. Several works sharing this view include Overlay MAC [20] and middleware-based control [14] [13]. These systems, however, do not integrate the CPU scheduling into the real-time system and cannot provide a complete end-to-end delay guarantee. iDSRT does this by integrating all three important schedulers: task scheduler, packet scheduler and node scheduler.

In the third category, essentially, previous work has shown the need for integrating the task scheduler and the network scheduler referred to as multi-resource coordination/reservation and scheduling problem [11] [12] [24] [22] [21]. In [22] [18], the approach is to allocate the resources such that the end-to-end delay can be guaranteed while optimizing the general resource utilization. Xu et al. [24] tries to provide best end-to-end QoS level for an application under the constraints of resource availability in wired networks. In [11] [12] end-to-end delay is achieved by assigning deadlines for each resource such that the number of future applications admitted is maximized. However, these works do not address the need for the *coordination among the nodes* because it considers the wired networks. In the wireless scenario, nodes share the wireless medium and thus need to coordinate with each other. The node scheduler is required and this motivates the need for the Coordinator. iDSRT addresses both issues. Thus, it works in the wireless scenario as shown in the paper and should work in the wireline scenario.

7 Conclusions

We have shown an integrated soft real-time scheduling framework, i.e. multi-resource allocation and scheduling for periodic soft-real-time tasks in wireless LAN environment. This is the first integrated system that considers both scheduling and coordination of three important entities in WLAN: the RT tasks, the RT packets and the nodes that share the wireless medium. The result of iDSRT

clearly show that augmented Linux and 802.11 WLAN technologies are feasible for critical infrastructures such as PowerGrid SCADA systems and can yield delay and loss guarantees currently only achievable over the wired network with modified general purpose kernels. We believe that iDSRT allows an easy deployment of general purpose hardware and software in PowerGrid substation, while preserving a major requirement of the real-time guarantees.

References

1. General electric wireless SCADA/Telemetry networking, <http://www.microwavedata.com/applications/scada/>
2. SEL-3022 wireless encrypting transceiver, <http://www.selinc.com/sel-3022.htm>
3. IEEE P1777/D1: Draft recommended practice for using wireless data communications in power system operations (February 2007)
4. IEEE Standard 1646: Communication delivery time performance requirements for electric power substation automation (September 2004)
5. IEEE standard 802.11e (September 2004), <http://standards.ieee.org/getieee802/802.11.html>
6. Ayers, Yodaiken, B.V.: Introducing real-time linux. *Linux Journal* 1997(34es), 5 (1997)
7. Caccamo, M., Zhang, L.Y., Sha, L., Buttazzo, G.: An implicit prioritized access protocol for wireless sensor networks. In: *Proceedings of the IEEE Real-Time Systems Symposium, RTSS (2002)*
8. Crenshaw, T.L., Hoke, S., Tirumala, A., Caccamo, M.: Robust implicit EDF: A wireless MAC protocol for collaborative real-time systems. *Transaction on Embedded Computing System* (2007)
9. Ergen, M., Duke Lee, R.S., Varaiya, P.: WTRP: Wireless token ring protocol. *IEEE Transaction on Vehicular Technology* (2004)
10. Gleixner, T., Molnar, I.: ktimers subsystem, <http://lwn.net/articles/152363/>
11. Gopalan, K., cker Chiueh, T.: Multi-resource allocation and scheduling for periodic soft real-time applications. In: *Proceedings of ACM/SPIE Multimedia Computing and Networking (2002)*
12. Gopalan, K., Kang, K.-D.: Coordinated allocation and scheduling of multiple resources in real-time operating systems. In: *Proceedings of Workshop on Operating Systems Platforms for Embedded Real-Time Applications, OSPERT (2007)*
13. He, W., Nahrstedt, K.: Impact of upper layer adaptation on end-to-end delay management in wireless ad hoc networks. In: *12th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS (2006)*
14. He, W., Nguyen, H., Nahrstedt, K.: Experimental validation of middleware-based QoS control in 802.11 wireless networks. In: *3rd International Conference on Broadband Communications, Networks, and Systems, BROADNETs (2006)*
15. hua Chu, H.: CPU Service Classes: A Soft Real Time Framework for Multimedia Applications. PhD thesis, UIUC (1999)
16. Jones, M., Alessandro, J., Paul, F., Leach, J., RoOu, D., RoOu, M.: An overview of the rialto realtime architecture (1996)
17. Liu, C.L., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM* 20(1), 46–61 (1973)
18. Nahrstedt, K., hua Chu, H., Narayan, S.: QoS-aware resource management for distributed multimedia applications. *Journal on High-Speed Networking, Special Issue on Multimedia Networking (1998)*

19. Nieh, J., Lam, M.S.: The design of SMART: A scheduler for multimedia applications. Technical Report CSL-TR-96-697 (1996)
20. Rao, A., Stoica, I.: An overlay MAC layer for 802.11 networks. In: 3rd International Conference on Mobile Systems, Applications, and Services (2005)
21. Shankaran, N., Koutsoukos, X.D., Schmidt, D.C., Xue, Y., Lu, C.: Hierarchical control of multiple resources in distributed real-time and embedded systems. In: Euromicro Conference on Real-time systems (2006)
22. Sourav Ghosh, J.H., Rajkumar, R., Lehoczky, J.: Integrated resource management and scheduling with multi-resource constraints. In: Proceedings of the IEEE Real-Time Systems Symposium, RTSS (2004)
23. Srinivasan, B., Pather, S., Hill, R., Ansari, F., Niehaus, D.: A firm real-time system implementation using commercial off-the-shelf hardware and free software. In: Proceedings of the Fourth IEEE Real-Time Technology and Applications Symposium, RTAS (1998)
24. Xu, D., Nahrstedt, K., Viswanathan, A., Wichadakul, D.: Qos and contention-aware multi-resource reservation. In: IEEE International Symposium on High Performance Distributed Computing, HDPC (2000)
25. Yang, Y., Kravets, R.: Achieving delay guarantees in ad hoc networks through dynamic contention window adaptation. In: IEEE Conference on Computer Communication, INFOCOM (2006)
26. Yang, Y., Wang, J., Kravets, R.: Distributed optimal contention window control for elastic traffic in wireless LANs. In: IEEE Conference on Computer Communication, INFOCOM (2005)
27. Yuan, W.: GRACE-OS: An Energy-Efficient Mobile Multimedia Operating System. PhD thesis, UIUC (2004)

Cell Breathing Based on Supply-Demand Model in Overlapping WLAN Cells*

Shengling Wang¹, Yong Cui¹, Ke Xu¹, Sajal K. Das², Jianping Wu¹,
and Yanping Xiao¹

¹ Dept. of Computer Science and Technology, Tsinghua University
Beijing, China

² University of Texas at Arlington

{slwang, cy, xuke, ypxiao}@csnet1.cs.tsinghua.edu.cn,
jianping@cernet.edu.cn, das@cse.uta.edu

Abstract. Introducing cell breathing in cellular networks into wireless local area networks (WLANs) for load balancing is beneficial since no special modification of clients. However, fairness and effectiveness is quite challenging in cell breathing. In this paper, a supply-demand model (SDM) based on cell breathing technique is proposed to allocate continuous or discrete power to APs for fair and effective load balancing. SDM classifies the beacon power of an AP into two kinds: the demand power and the supply power. The former is the ideal power that an AP is supposed to have while the latter is the power the AP actually transmits. Finding the deterministic global optimal solution, SDM makes the demand and supply power as equal as possible and the load on APs balanced. Because SDM does not need multiple iterations to compute the optimum, it can avoid frequent user handoffs resulting from frequent power change. Finally, SDM is extended to support a broader range of load definition and the generalized relationship between beacon power and load. The simulation results show the proposed scheme is fair for realizing load balancing and effective for improving throughput.

Keywords: Load balancing, cell breathing, power assignment.

1 Introduction

In WLANs, load imbalance usually comes up, which incurs two problems: lower network throughput and worse quality of service (QoS). In the carrier sense multiple access (CSMA)-based WLANs, all clients have equal rights to access the wireless medium. Thus, in a congested AP, there exists a higher probability that multiple clients access the wireless medium at the same time, resulting in a large number of transmission collisions. As a consequence, more bandwidth will be consumed for retransmissions, leading to lower network throughput. Moreover, longer backoff periods are needed to avoid collisions, thus resulting in longer transmission delay.

* This work is supported by NSFC Project (no. 60873252), International S&T Cooperation Program of China (no.2008DFA11630), National Major Basic Research Program of China (no. 2007CB307105, 2009CB320501, 2009CB320503), postdoctoral foundation (no.20090450391).

However, the IEEE 802.11 standard, the protocol of WLANs, does not provide any method for load balancing. To make up this deficiency, both industry and academia have proposed some solutions [1-10], most of which require clients to select APs based on not only the received signal strength indicator (RSSI) but also the load on the APs. This requirement needs clients to have special software/hardware supports. In real networks, WLAN clients could be heterogeneous with different AP selection policies, which make it hard to cooperate with APs for load balancing due to proprietary schemes of different device providers.

To solve the above problems, some researchers [11-13] introduce the concept of cell breathing in code division multiple access (CDMA) network into WLANs for load balancing. In fact, cell breathing is a side effect in CDMA networks in the sense that the coverage and capacity of a CDMA cell are reduced with the increase of user number. However, it can be a load balancing technique in WLANs if some optimal strategies are applied.

When cell breathing method is used in WLANs, if an AP is heavily loaded, it will reduce the power of beacons to shrink its coverage area for reducing the serving of new clients as shown in Fig. 1(a); if the AP is lightly loaded, it will increase the power of beacons to expand its coverage area for attracting new clients as shown in Fig. 1 (b).

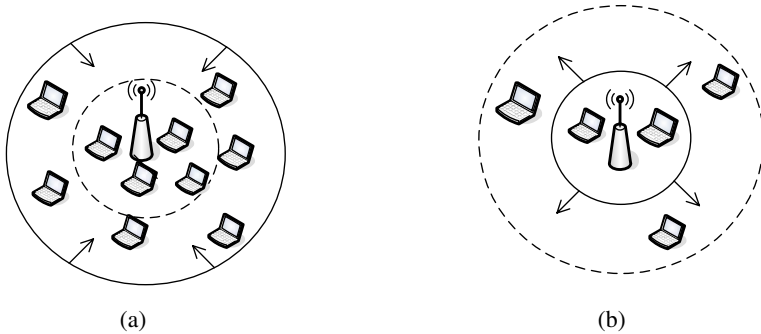


Fig. 1. Example of cell breathing (a) AP is heavily-loaded (b) AP is lightly-loaded

Reducing data transmission power of an AP will degrade the channel quality of all its associated users, not only those who tend to shift to other APs, but also those who still associate with the current AP. To solve this problem, cell breathing technique separates the transmission power of data and beacons. It only adjusts the power of beacons because the beacon power only affects the cell dimension and has no impact on the loss rate and transmission latency of data packets.

In cell breathing method, clients are not required any modification. They select an AP only according to the default mode of IEEE 802.11. Therefore, it can be realized easily and has a broader application prospect.

Although conceptually simple, implementing cell breathing is surprisingly challenging. We explain this problem through the following *Example*.

Example: Consider a WLAN with three APs, *A*, *B* and *C* and ten users. For simplicity, we assume that all users generate the same traffic load, and all APs have the same bit rate and data transmission power. At the beginning, due to the extremely different beacon power, the network load is quite imbalanced as shown in Fig.2 (a). To solve this problem, we use the cell breathing method that increases the beacon power of APs *A* and *C*, and decreases that of AP *B*. From the viewpoint of load balancing, we can obtain two optimal results as shown in Figs. 2(b) and (c). Both client-to-AP mappings can realize load balancing, but their throughput is different.

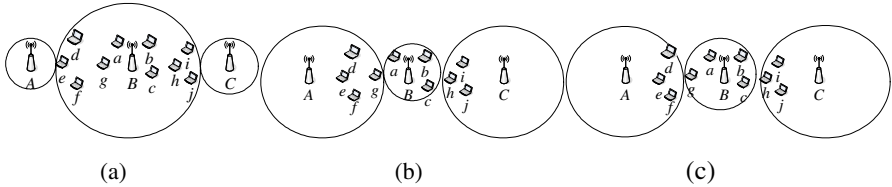


Fig. 2. Challenges in Cell Breathing

The difference between Figs. 2(b) and (c) is whether user *g* accesses AP *B* or *A*, which leads to different throughput. When each AP's data transmission power is same and without considering other factors, the throughput of a user is proportional to its received power [12] which is in turn inversely proportional to the distance between the user and AP [14]. Because user *g* is nearer to AP *B*, the throughput when it accesses *B* is higher than accesses AP *A*.

In reality, WLANs offer more complicated scenarios. So our challenge is to propose an effective cell breathing method to find an optimal power assignment that achieves load balancing while providing higher throughput. In addition, another challenge is to avoid the frequent power change, because it will induce non-negligible handoff latency.

This paper proposes a cell breathing scheme based on the supply-demand model (SDM). It classifies the beacon power of an AP into two kinds: the demand power and the supply power. The former is the ideal power that an AP is supposed to have while the latter is the power the AP actually transmits. SDM makes the demand and supply power as equal as possible while the load on APs balanced. The proposed scheme inherits the advantages of cell breathing method, and may have the following potential contributions:

- 1) It finds the deterministic global optimal mapping between users and APs instead of relying on local optimization heuristics. More importantly, it does not need multiple iterations for finding the optimum, avoiding frequent user handoffs caused by frequent power change.
- 2) It supports a broad range of load definition besides the user number.
- 3) It is applicable to both continuous power and discrete power assignment.

The simulation results show that our SDM-based cell breathing scheme is fair for realizing load balancing and effective for improving throughput.

The rest of paper is organized as follows: related works are shown in Section 2 while the problem formulation is given in Section 3. SDM are presented in Section 4,

and how to assign power through SDM-based cell breathing method is described in Section 5, which is extended in Section 6. The performance evaluation is discussed in Section 7 while conclusions are offered in Section 8.

2 Related Works

Load balancing in WLANs is an important issue that has attracted significant attentions from both industry and academia. From industry perspective, several vendors [1-4] have supported load balancing in their WLAN products. For example, Cisco's load balancing solution [1] is based on the client number, signal strength and bit error rates. D-link [2], Proxim [3] and Airflow [4] take into account the actual AP traffic in their solutions. In these proprietary products, APs need to broadcast their load information to clients through beacons, and each client is required to select the AP with the lightest load.

From academic perspective, several algorithms have been proposed to distribute the load based on different load metrics, such as the client number [5], the capacity available for a new station that uses the fastest modulation (AAC) [13], the aggregated downlink and uplink traffic [6], the channel idle time [7], the channel utilization estimate (CUE) [8]. Balachandran et al. [9] proposed to select an AP in light of not only its RSSI but also the minimal bandwidth that the AP can offer. Bejerano et al [10] proved the strong correlation between fairness and load balancing, and used load balancing techniques to obtain near-optimal max-min fair bandwidth allocation.

All the above schemes require the clients to have special supports for AP selection. To avoid revising clients, some researchers [11-13] introduced the concept of cell breathing in CDMA networks into WLANs for load balancing.

Bejerano and Han [11] presented a cell breathing method for load balancing in WLANs, which is based on two algorithms: one minimizes the load of the most congested AP(s), and the other produces an optimal min-max load balancing solution. These methods can maximize network throughput while providing fairness. However, if only the currently associated user set and each user's load are known, their algorithms need multiple iterations to converge to an optimum. Because each iteration will actually change the beacon power which may trigger some clients to move among the coverage areas of APs from time to time, this method may induce non-negligible handoff latency up to 1.3 seconds [7].

Bahl et al. [12] proposed another cell breathing scheme. Different from [11], this scheme does not need multiple iterations. Once the topology of APs and clients is given, an optimal mapping of clients to APs can be found with the help of linear programming. However, their method needs to know the distance between each AP-client pair, hence increasing the algorithm complexity.

Garcia et al. [13] adopted a cell breathing method to adjust the cell size according to the signal to noise ratio (SNR) received by each client in the AP's coverage area. To obtain the SNR information, their scheme needs the help of some new standards such as the IEEE 802.11k or h.

3 Problem Formulation

In this section, the problem formulation is given. As described above, cell breathing method separates the transmission power of data and beacons. So we call the beacon signal strength indicator a user receives as RSSI-B while the data signal strength indicator a user receives as RSSI-D.

Due to without any special modification of clients, default WLAN users select APs only according to RSSI-B of APs. However, RSSI-D of APs that they receive actually affects the data link quality. In our scheme, to reflect the relationship between AP and users' data link quality, users are classified into two types:

Definition 1: A primary user of AP i ($i = 1, 2, \dots, N$) is defined as a user whose RSSI-D sensed from AP i is the loudest, where N is the maximum AP number in the network.

Definition 2: A handoff user of AP i is defined as a primary user of other APs, who currently associates with AP i ($i = 1, 2, \dots, N$).

Obviously, a user of AP i ($i = 1, 2, \dots, N$) is either a primary user or a handoff user. In our scheme, time is divided into equal length intervals. The number of primary users belonging to each AP at any time interval makes up the primary user vector, which is defined as:

Definition 3: The primary user vector $P(k) = [p_1(k) \ p_2(k) \ \dots \ p_N(k)]$, where $p_i(k)$ is the primary user number of AP i ($i = 1, 2, \dots, N$) during the k^{th} time interval, $k = 1, 2, \dots$.

In addition, despite the user type, the number of users connecting to each AP at any time interval makes up the user vector, whose definition is:

Definition 4: The user vector $U(k) = [u_1(k) \ u_2(k) \ \dots \ u_N(k)]$, where $u_i(k)$ is the number of users that associate with AP i ($i = 1, 2, \dots, N$) during the k^{th} time interval, $k = 1, 2, \dots$.

Obviously, both $\sum_{i=1}^N p_i(k)$ and $\sum_{i=1}^N u_i(k)$ are the total user number in the network during the k^{th} time interval.

For the sake of simplicity, we assume that the AP deployment ensures a high degree of overlaps among the range of adjacent APs as in [11]. And then our problem is when the primary user vector is given, how to find the optimal user vector to make each user associate with the AP providing the data link quality as good as possible while guaranteeing the load balancing on APs. To solve the problem, we propose SDM as described in the next section.

4 Supply-Demand Model

4.1 Formulations of Demand and Supply Power

SDM classifies the beacon power of an AP into two kinds: the demand power and the supply power. The former is the ideal power that an AP is supposed to have while the latter is the power the AP actually transmits.

According to the definition of primary users, an AP should serve its primary users as many as possible to provide better data link quality to users. At the same time, when the AP is light-loaded while its neighbors are congested, it should serve some users of its neighbors to relieve their load. So, an AP should have enough beacon power to attract the primary and handoff users.

According to [12], the larger the beacon power is, the more users that request to associate with an AP. Here, we assume the relationship between the associated user number and the beacon power to be linear correlation. Actually, our model is still workable when this assumption is replaced by other relationship as described in Section 6. Therefore, the demand power of AP i at the k^{th} time interval, denoted as $d_i(k)$, can be formulated as follow:

$$d_i(k) = \alpha p_i(k) + f_i + \sum_{j=1}^N q_{ij} \cdot (d_j(k-1) - s_j(k-1)) \quad (i = 1, 2, \dots, N) \quad (1)$$

Formula (1) includes two parts. The first part $\alpha p_i(k) + f_i$ is the power of AP i needed to attract the primary users at the k^{th} time interval. This part is proportional to the primary user number. Here, $\alpha > 0$ is the proportionality coefficient and f_i is a constant reflecting the external factors (e.g., geomorphology, buildings or trees) that affect AP i to provide power for attracting users.

The second part $\sum_{j=1}^N q_{ij} \cdot (d_j(k-1) - s_j(k-1))$ is the power provided by AP i to attract handoff users at the k^{th} time interval, where $d_j(k-1)$ and $s_j(k-1)$ are the demand power and supply power of AP j at the $(k-1)^{\text{th}}$ time interval respectively. This part is closely related to the power provided by the neighbor APs at the $(k-1)^{\text{th}}$ time interval. During the $(k-1)^{\text{th}}$ time interval, if the demand power in the neighbors of AP i is larger than the supply power, i.e. $d_j(k-1) > s_j(k-1)$, their primary users will be compelled to shift. As a result, AP i should increase its power to attract these handoff users, and vice versa. In Formula (1), q_{ij} is the power-impact factor of AP j to AP i . The power-impact factors of all APs constitute the power-impact factor matrix, whose definition is:

Definition 5: The power-impact factor matrix $Q = \{q_{ij}\}_{N \times N}$, where q_{ij} is the power-impact factor of AP j to AP i . In particular, q_{ij} is the fraction of primary users in AP j who shift to AP i due to power deficiency in AP j resulting from its congestion; or q_{ij} is the fraction of handoff users in AP j who comes from AP i due to the excessive power in AP j because of its light load. Note $q_{ij} = 0$, if AP j is not the neighbor of AP i or $i=j$. Apparently, $\sum_{i=1}^N q_{ij} = 1 \quad (j = 1, 2, \dots, N)$.

In our scheme, a WLAN client requires neither special support nor change in the standard. It chooses APs only according to the received RSSI-B. As a result, q_{ij} is

only related to those factors that affect the RSSI-B received by users, such as the distance between APs, the geomorphology, buildings, trees and other obstacles.

Likewise, we also assume that the supply power of AP i during the k^{th} time interval, denoted as $s_i(k)$, is proportional to the number of its associated users, $u_i(k)$. So $s_i(k)$ can be formulated as follow:

$$s_i(k) = \alpha u_i(k) + f_i \quad (i = 1, 2, \dots, N; k = 1, 2, \dots) \quad (2)$$

4.2 Optimal User Vector Solution

According to Formulas (1) and (2), we obtain the following definition:

Definition 6: The supply-demand deficit vector is $M(k) = [m_1(k) \ m_2(k) \ \dots \ m_N(k)]$, where $m_i(k)$ is the supply-demand deficit of power in AP i during the k^{th} time interval. With respect to Formulas (1) and (2), $m_i(k)$ ($i = 1, 2, \dots, N; k = 1, 2, \dots$) can be calculated as:

$$m_i(k) = d_i(k) - s_i(k) = \alpha p_i(k) + \sum_{j=1}^N q_{ij} \cdot (d_j(k-1) - s_j(k-1)) - \alpha u_i(k) \quad (3)$$

$$(i = 1, 2, \dots, N; k = 1, 2, \dots)$$

Formula (3) can be rewritten using vector form:

$$M(k) = \alpha P(k) + M(k-1) \cdot Q - \alpha U(k) \quad (4)$$

Aiming at our problem described in Section 3, we need to find $U(k)$ that realizes two objectives: (i) minimizing the supply-demand deficit of power to make the supply power and demand power as equal as possible; (ii) making the load of each AP approach to the mean load of its neighbors to realize load balancing. So we can get the following formula:

$$\begin{aligned} & \min (1-\varphi)M(k)M(k)^T + \varphi(U(k) - \bar{U}(k))(U(k) - \bar{U}(k))^T \\ & \text{s.t. } U(k)A = P(k)A \\ & A = \overbrace{[1 \ 1 \ \dots \ 1]}^N \end{aligned} \quad (5)$$

In Formula (5), the first part guarantees the objective (i) while the second part guarantees the objective (ii). $\varphi \in [0, 1]$ is the weight coefficient.

$\bar{U}(k) = [\bar{u}_1(k) \ \bar{u}_2(k) \ \dots \ \bar{u}_N(k)]$ is the mean value vector. $\bar{u}_i(k) = \sum_{j=1}^{A_i} p_j(k) / A_i$, where A_i is the neighbor number of AP i ($i = 1, 2, \dots, N$). In addition, the constraint condition guarantees $\sum_{i=1}^N p_i(k) = \sum_{i=1}^N u_i(k)$ = the user number in the network. By taking the derivative of Formula (5), the optimum is obtained as:

$$U(k) = \frac{\alpha^2(1-\varphi)(P(k) + M(k-1)(\bar{Q} + \bar{Q})) + \varphi\bar{U}(k)}{\alpha^2(1-\varphi) + \varphi} \quad (6)$$

where $\bar{Q} = \begin{bmatrix} q_1/N & q_1/N & \dots & q_1/N \\ q_2/N & q_2/N & \dots & q_2/N \\ \dots & \dots & \dots & \dots \\ q_{N-1}/N & q_{N-1}/N & \dots & q_{N-1}/N \\ q_N/N & q_N/N & \dots & q_N/N \end{bmatrix}$ and $q_i = \sum_{j=1}^N q_{ij} \quad i = 1, 2, \dots, N$.

5 Power Assignment Based On SDM

5.1 Continuous and Discrete Power Assignment

In actual scenario, some APs can adjust their power to any values, while others only to certain discrete values. In this section, we explain how to use SDM to realize cell breathing when the power of AP is continuously adjustable and only a set of discrete values as shown in Figs. 3 and 4 respectively.

```

1 initialize  $D(0), S(0)$  and time slot sequence number  $k \leftarrow 1$ ;
2 IF(a time interval terminates)
2.1 obtain the neighbors' supply-demand deficit in power;
2.2 estimate  $P(k)$  and compute  $D(k), U(k), S(k)$ ;
2.3 FOR ( $j = 1; j \leq N; j++$ )
2.3.1 IF( $u_j(k) > u_j^{MAX}$ )  $u_j(k) \leftarrow u_j^{MAX}; s_j(k) \leftarrow \alpha u_j(k) + f_j$ ;
2.3.2 IF( $s_j(k) > Pow_j^{MAX}$ )  $s_j(k) \leftarrow Pow_j^{MAX}$ ;
2.4  $k \leftarrow k + 1$ ;
    
```

Fig. 3. Cell Breathing for Continuous Power

```

1 initialize  $D(0), S(0)$  and time slot sequence number  $k \leftarrow 1$ ;
2 IF (a time interval terminates)
2.1 obtain the neighbors' supply-demand deficit in power;
2.2 estimate  $P(k)$  and compute  $D(k), U(k), S(k)$ ;
2.3 FOR( $j = 1; j \leq N; j++$ )
2.3.1 IF( $u_j(k) > u_j^{MAX}$ )  $u_j(k) \leftarrow u_j^{MAX}; s_j(k) \leftarrow \alpha u_j(k) + f_j$ ;
2.3.2 FOR each power in [ $Pow_j^1 \quad Pow_j^2 \quad \dots \quad Pow_j^{MAX}$ ];
2.3.2.1 find a power level which is closest to  $s_j(k)$ ;
2.4  $k \leftarrow k + 1$ ;
    
```

Fig. 4. Cell Breathing for Discrete Power Level

In both figures, u_j^{MAX} is the maximum user number that AP j can serve. In Fig.3, the power of AP j can change from 0 to Pow_j^{MAX} ; however, in Fig. 4, the power level of AP j is a set of discrete values, say $[Pow_j^1 \quad Pow_j^2 \quad \dots \quad Pow_j^{MAX}]$, ($j = 1, 2, \dots, N$).

5.2 Some Key Parameters

From the figures, each AP needs to initialize its demand power and supply power. To obtain the primary users, we set both $d_i(0)$ and $s_i(0)$ of beacons equal to the data transmission power of AP i .

When each AP transmits beacons with the same power as that of data, the beacons' RSSI is the same as that of data, i.e. $RSSI-D=RSSI-B$. Because cell breathing does not need any modification of clients, RSSI-D may be the most easily obtained parameter reflecting data link quality. Thus, when $RSSI-D=RSSI-B$, the strongest RSSI-B received by a user from an AP means that the AP can provide the best data link quality to the user. Because the primary users of an AP are those who can obtain the loudest RSSI-D, when $d_i(0)$ and $s_i(0)$ of beacons equal to the data transmission power of AP i , the users who request to associate with AP i according to RSSI-B are just its primary users.

In addition, except the first time interval, at the beginning of other time intervals, our scheme needs to estimate the number of primary users in AP i . For this purpose, we use the following formula:

$$p_i(k) = p_i(k-1) + p_{ni}(k) - p_{li}(k) \quad (i = 1, 2, \dots, N) \tag{7}$$

In Formula (7), $p_{ni}(k)$ is the number of newly arrival primary users in AP i during the k^{th} time interval and $p_{li}(k)$ is the number of primary users who leave AP i during that interval. Now $p_{ni}(k)$ and $p_{li}(k)$ can be estimated from some empirical data, while $p_i(k-1)$ can be calculated in terms of $u_i(k-1)$, and the power supply-demand deficit of AP i and its neighbors.

When the power of AP i is enough at the $(k-1)^{th}$ time interval, the power of its neighbors have two statuses: (i) the power of all neighbors is enough; (ii) the power of a part of or all neighbors is shortage. In status (i), evidently, $p_i(k-1) = u_i(k-1)$; in status (ii), some users of AP j ($j \in C_i$) will roam to the range of its neighbors, where C_i is the set of AP i 's neighbors with power shortage. We denote the number of such users as $H_j(k-1)$. Because the power of AP i is lightly loaded, it should serve a fraction of users coming from AP j . In light of the definition of power-impact factor, we know the number of handoff users from AP j is $q_{ij}H_j(k-1)$. As a result, in status

(ii), $p_i(k-1) = u_i(k-1) - \sum_{j=1}^{C_i} q_{ij}H_j(k-1)$. When the power of AP i falls short during the $(k-1)^{th}$ time interval, the primary users of this AP will shift to other APs. In this scenario, $p_i(k-1) = u_i(k-1) + H_i(k-1)$. Here, $H_i(k-1)$ ($i = 1, 2, \dots, N$) can be

obtained according to the power supply-demand deficit of AP i and its neighbors, and the relationship between power and the number of users.

6 SDM Extension

As shown in Formulas (1) and (2), SDM has two main assumptions: (a) the user number is linearly direct proportional to the beacon power transmitted by an AP; (b) the load metric is user number. In fact, our model can work without the above assumptions. In this section, we extend SDM to a broader application range. The extended demand power is defined as follow:

$$d_i(k) = \overline{d_i(k)} + \sum_{j=1}^N q_{ij} \cdot (d_j(k-1) - s_j(k-1)) \quad (8)$$

where $\overline{d_i(k)}$ is defined as the primary demand power, formulated as:

$$\overline{d_i(k)} = G_1(\overline{L_i}, f_i) \quad (i=1, 2, \dots, N) \quad (9)$$

In Formula (9), the definition of f_i is the same as that in Section 4. $\overline{L_i}$ is the load which has two kinds: the *primary* load and the minimum expected load, as defined as follow:

Definition 7: The *primary load* of AP i is the load that can obtain the best QoS when it is served by AP i ($i=1, 2, \dots, N$) instead of other APs.

Apparently, the primary user is a special case of the primary load.

Definition 8: The *minimum expected load* of AP i is the minimum load that AP i ($i=1, 2, \dots, N$) expects to serve.

The extended supply power is given in Formula (10), where L_i is the actual load served by AP i .

$$s_i(k) = G_2(L_i, f_i) \quad (i=1, 2, \dots, N) \quad (10)$$

Formulas (9) and (10) show that $\overline{d_i(k)}$ is the function of $\overline{L_i}$ and f_i , and $s_i(k)$ is the function of L_i and f_i . The form of $G_1(\bullet)$ and $G_2(\bullet)$ can be fitted by the empirical data about the relationship between power and load.

Besides generalizing the relationship between power and load, another key extension is load definition. The load in the extended SDM is not tied to the user number, but supports a broad range of load definition. For example, the load can be traffic, the channel idle time [7], CUE [8]. Correspondingly, $\overline{L_i}$ can be the primary traffic (generated by primary user), the minimum expected channel idle time, the minimum expected CUE. In the extended model, APs can get its load through real-time measurement, or through users by means of some new standards, such as the IEEE 802.11 k or h.

After extending the definitions of demand power and supply power, the optimal mapping between the APs and users can be solved through the method introduced in Sections 4 and 5.

7 Performance Evaluation

In this section, we compare the performance of our scheme with the default WLAN AP selection scheme. As described above, default WLAN users select APs only according to their RSSI-B. Although our scheme has the same AP selection criterion, the APs adjust their beacon power dynamically according to the network load.

In our simulation, we randomly place 20 APs in a 500m×500m area. The distributions of clients have two scenarios: random pattern and hotspot pattern. To realize these distributions, we number all APs and appoint APs 19 and 20 as the serving APs in hotspot areas. The probability that clients are nearest to these two APs is P_{hot} , while the probability that other clients are nearest to APs 1-18 are equal, i.e. $(1-P_{hot})/18$. Therefore, the user-distribution is in random pattern when the fraction of primary users in each APs is $1/20$, while it is in hotspot pattern when $P_{hot} > 0.1$.

The simulation sets the proportional coefficient $\alpha = 1$ and the weight coefficient $\varphi = 0.6$. For simplicity, we assume the constant reflecting the external factors $f_i (i = 1, 2, \dots, N) = 0$ and let the power-impact factor only relate to the distance among APs. In addition, we assume that all APs have the same transmission power of data. As a result, the primary user of one AP is the user whose distance to this AP is the closest. Each element in the power-impact factor matrix can be calculated by Formula (11), where $dis(i, j)$ is the distance between AP i and j .

$$q_{ij} = \begin{cases} \frac{dis(i, j)}{\sum_{k \in A_j} dis(j, k)} & j \in A_i \\ 0 & otherwise \end{cases} \quad i = 1, 2, \dots, N \quad (11)$$

We compare SDM with default WLAN scheme in terms of load balancing and the normalized system throughput, T_N , which is defined as the fraction of time that the channel is used to successfully transmit payload bits. T_N can be calculated using Formula (12) [15].

$$T_N = \frac{P_s P_{tr} E[P]}{(1 - P_{tr})\sigma + P_s P_{tr} T_s + P_{tr} (1 - P_s) T_c} \quad (12)$$

where P_{tr} and P_s can be calculated as:

$$P_{tr} = 1 - (1 - \tau)^n \quad (13)$$

$$P_s = n\tau(1 - \tau)^{n-1} / P_{tr} \quad (14)$$

$$\tau = \begin{cases} 2(1-2\rho)/((1-2\rho)(W+1)+\rho W(1-(2\rho)^m)) & \rho \neq 0.5 \\ 2/(W+1) & \rho = 0.5 \end{cases} \quad (15)$$

Here P_{tr} denotes the probability that there is at least one transmission in a given slot and P_s denotes the probability that the transmission is successful; τ is the probability that a user transmits data; n is the number of clients that associate with an AP. The meanings and values of other parameters are shown in Table.1 [15].

Table 1. Simulation Parameters

Para.	Description	Value
$E[P]$	Average packet payload size (measured in $50 \mu s$ slot time units)	163.68 slot time units
σ	Duration of an empty slot time	$50 \mu s$
T_s	Average time the channel is sensed busy because of a successful transmission	179.64 slot time units
T_c	Average time the channel is sensed busy during a collision	174.26 slot time units
ρ	Conditional collision probability	$0.6n$
W	Minimum contention window	128
m	Maximum backoff stage	5

In addition, for the sake of simplicity, we assume the data transmission power and the frequency of APs are set reasonably so that the interference among APs can be ignored.

Figs. 5, 6 and 7 show the load on the APs in SDM and default WLAN scheme when the total user number in the network is 30, 60 and 90 respectively. The load in the simulation is measured by the number of users. In Fig.5, $P_{hot} = 0.1$, which means that the distribution of users is in random pattern. In Figs.6 and 7, $P_{hot} = 0.3$ and 0.5

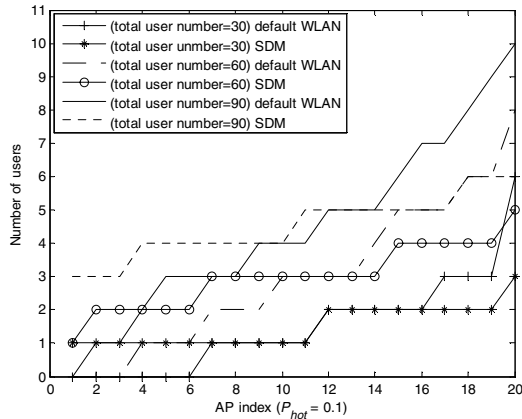


Fig. 5. Load on APs ($P_{hot} = 0.1$)

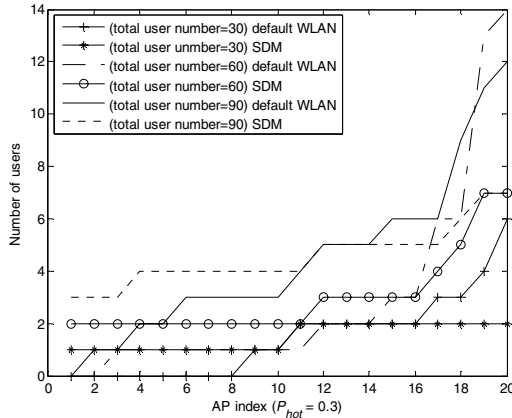


Fig. 6. Load on APs ($P_{hot} = 0.3$)

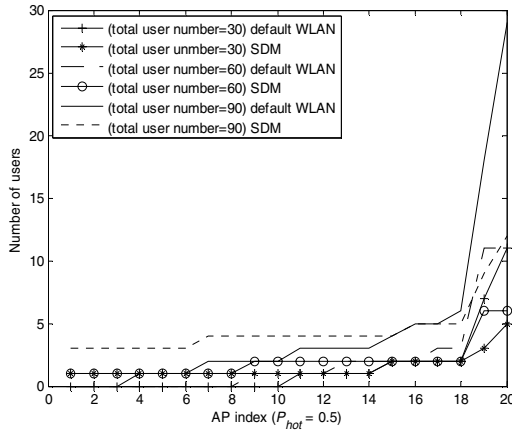


Fig. 7. Load on APs ($P_{hot} = 0.5$)

respectively, implying that the distributions of users are in hotspot pattern and the probabilities that users are closest to APs 19 and 20 are 30% and 50%. Note that the APs are sorted by their load in increasing order.

Fig. 8 shows how the sum of normalized system throughput of all APs changes with the total number of users in the network. It can be seen that the throughput increases as the increase of total user number. Moreover, the throughput in our method is greater than that in the default WLAN scheme. The reason is that the default WLAN users prefer to connect to the nearest APs. As a consequence, when many clients are nearest to one AP, they do not connect to other APs even though this AP is heavily loaded. In the CSMA-based WLAN, the more users converge to one AP, the larger is the transmission collisions, which leads to lower throughput. While in our method, the near-average load in each AP reduces the transmission collisions, thus enhancing the overall throughput.

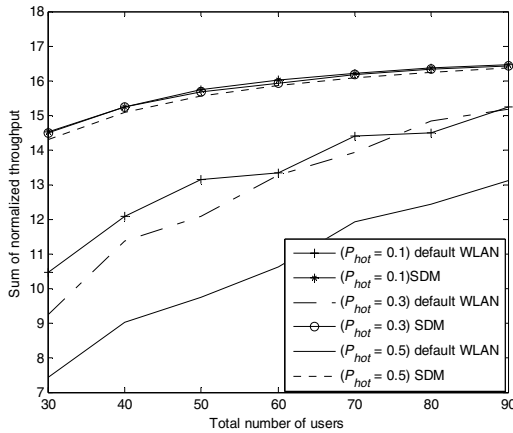


Fig. 8. Throughput comparison

8 Conclusion

We propose a cell breathing scheme based on SDM for continuous and discrete power assignment. It finds an optimal mapping between users and APs for providing better data link quality to users and realizing load balancing. To this end, our scheme makes the demand power and supply power as equal as possible and the load on the APs balanced. Finally, we extend SDM by generalizing the relationship between power and load and defining the load in a broader sense. The simulation results demonstrate that our scheme is fair for realizing load balancing and effective for improving throughput.

References

1. Cisco system, Inc. Data sheet cisco aironet, 350 Series access points, <http://www.csico.com>
2. <http://www.dlink.com/products/?pid=303>
3. Proxim wireless networks. Data sheet Orinoco AP-600 access point, <http://www.proxim.com>
4. AirFlow networks, Inc. White paper: high availability for mission-critical WLANs, <http://www.airflownetworks.com>
5. Papanikos, I., Logothetis, M.: A study on dynamic load balance for IEEE 802.11b wireless LAN. In: Proc. COMCON (2001)
6. Velayos, H., Aleo, V., Karlsson, G.: Load balancing in overlapping wireless LAN cells. In: Proc. IEEE ICC 2004, pp. 3833–3836 (1998)
7. Guo, F., Chiueh, T.: Scalable and robust WLAN connectivity using access point array. In: Proc. International Conference on Dependable Systems and Networks (DSN), July 2005, pp. 288–297 (2005)

8. Sawma, G., Aib, I., Ben-El-Kezadri, R., Pujolle, G.: ALBA: An autonomic load balancing algorithm for IEEE 802.11 wireless networks. In: Proc. IEEE Network Operations and Management Symposium (NOMS 2008), Salvador, Bahia, April 2008, pp. 891–894 (2008)
9. Balachandran, A., Bahl, P., Voelker, G.M.: Hot-spot congestion relief and service guarantees in public-area wireless networks. SIGCOMM Computing Communication Rev. 32(1), 59 (2002)
10. Bejerano, Y., Han, S.-J., Li, L.E.: Fairness and load balancing in wireless LANs using association control. In: Proc. ACM Mobicom 2004, Philadelphia, PA, USA, pp. 315–329 (2004)
11. Bejerano, Y., Han, S.: Cell breathing techniques for load balancing in wireless LAN. In: Proc. IEEE INFOCOM 2006, Barcelona, Spain, April 2006, pp. 1–13 (2006)
12. Bahl, P., Hajiaghayi, M.T., et al.: Cell breathing in wireless LANs: algorithms and evaluation. IEEE Transaction on Mobile Computing 6(2), 164–178 (2007)
13. Garcia, E., Vidal, R., Paradells, J.: Cooperative load balancing in IEEE 802.11 networks with cell breathing. In: Proc. IEEE Symposium on Computers and Communications (ISCC 2008), Marrakech, September 2008, pp. 1133–1140 (2008)
14. Elmusrati, M., Koivo, H.: Centralized algorithm for the tradeoff between total throughput maximization and total power minimization in cellular systems. In: Proc. IEEE 58th Vehicular Technology Conference, VTC 2003-Fall, October 2003, pp. 1598–1602 (2003)
15. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)

Comparative Analysis of QoMIFA and Simple QoS

Esam Alnasouri, Ali Diab, Andreas Mitschele-Thiel, and Thomas Frenzel

Ilmenau University of Technology, Integrated HW/SW Systems Group
Gustav-Kirchhoff-Str. 1, 98693 Ilmenau, Germany
{esam.alnasouri, ali.diab, mitsch}@tu-ilmenau.de,
syslock@gmx.de

Abstract. This paper evaluates the performance of QoS-aware Mobile IP Fast Authentication Protocol (QoMIFA) compared to the well-known Simple QoS signaling protocol (Simple QoS) via simulation studies modeled in the ns2. The evaluation comprises the investigation of network load impact on both protocols with respect to the time required to reserve resources, number of dropped packets per handoff and number of packets sent as best-effort after the handoff is completed and until resources are reserved. Our simulation results show that QoMIFA is capable of achieving fast and smooth handoffs in addition to its capability of reserving resources very quickly. QoMIFA is approximately 97.75 % and 73.92 % faster than Simple QoS with respect to the average time required to reserve resources on downlink and uplink, respectively. It drops 79.63 % less on downlink and 46.6 % less on uplink and results in 98.40 % less packets sent on downlink as best-effort.

Keywords: QoS, RSVP, Mobility Management, MIFA.

1 Introduction

Ubiquitous access to information anywhere, anytime and anyhow is an important feature of future All-IP mobile communication networks, which will interconnect existing and future communication networks via a common IP core and provide higher data rates at lower costs. Achieving the goals of All-IP mobile communication networks forces network providers to overcome many challenges. A main challenge is how to guarantee suitable QoS for real-time services while moving from a point of attachment to another. In other words, how to achieve a fast re-reservation of resources during and after the handoff?

Current IETF standard used to support mobility in IP-based networks is Mobile IP in its two versions, version 4 (MIPv4) [1] and version 6 (MIPv6) [2]. Due to the long latency resulting from MIP, it is only applicable to support global mobility, termed as macro mobility as well. To avoid the problems of MIP and to satisfy the requirements of delay-sensitive applications, various solutions for mobility management have been developed. One of the well-known solutions is Mobile IP Fast Authentication Protocol (MIFA) [3], which is capable of achieving fast and smooth handoffs. MIFA lacks, however, of supporting QoS.

The ReSource reservation Protocol (RSVP) is the well-known solution introduced to support QoS [4]. It enables Internet applications to obtain different QoS for their data flows by reserving resources along the path between sender and receiver. RSVP lacks, however, of mobility support.

To provide QoS and mobility management simultaneously, many proposals proposed to couple between mobility and QoS solutions, so that handoffs are executed and simultaneously resources are reserved. QoMIFA [5] is one of the best proposals achieving such coupling. This protocol integrates between MIFA as a mobility management protocol and RSVP as a solution for QoS. The idea is to introduce a new object called “Mobility object” to RSVP control messages. This object is utilized to encapsulate MIFA control messages inside, which results in supporting QoS and mobility simultaneously.

This paper aims at providing a detailed performance evaluation of QoMIFA compared to Simple QoS with respect to the time required to reserve resources on downlink as well as uplink, number of dropped packets per handoff on each direction (uplink or downlink) and number of downlink packets sent as best-effort after the handoff is completed and until resources are reserved. The performance evaluation is achieved via simulation studies modeled in network simulator 2 (ns2) [6].

The rest of this paper is organized as follows: section 2 highlights the state of the art. Section 3 presents the performance evaluation of QoMIFA compared to Simple QoS. Finally, section 4 concludes with the main obtained results and future work.

2 State of the Art

The schemes coupling between QoS and mobility solutions can be broadly classified into three categories, namely hard coupled, loose coupled and hybrid coupled solutions [7]. Hard coupled solutions attempt to extend existing mobility management or QoS protocols to simultaneously support both. Well-known example is the Wireless Lightweight Reservation Protocol (WLRP) [8]. In contrast to hard coupled approaches, loose coupled solutions separate between the protocols managing mobility and those providing QoS. However, changes occurring in one of them force performing some actions in the other. Well-known examples are Localize RSVP (LRSVP) [9], Mobile RSVP (MRSVP) [10], Hierarchical Mobile RSVP (HMRSVP) [11], Simple QoS Signaling Protocol for Mobile Hosts in the Integrated Services Internet [12], etc. Finally, hybrid coupled solutions aim at reducing the signaling burden resulting from sending mobility and QoS control messages by extending QoS control messages to include signaling for mobility or vice versa. QoMIFA is a well-known example.

WLRP utilizes RSVP to reserve resources. The Mobile Node (MN) periodically transmits reports including information about beacons received to the serving Base Station (BS), which determines based on these reports the candidate BSs the MN may move to. Following this, the serving BS requests resources for the MN to be reserved passively in each candidate BS. This ensures QoS guarantee for the MN after the handoff.

The main objective of LRSVP is to localize RSVP reservation in an access network by introducing a new proxy to split the RSVP session into two sessions. The first is between the Corresponding Node (CN) and proxy, while the second is between the proxy and MN. LRSVP introduces two new control messages to the messages known from RSVP, namely a PATH Request and PATH Request Tear message. The two messages are used to accelerate the reservation of resources on the new path and the release of resources on the old path.

MRSVP extends RSVP to understand mobility. It distinguishes between two types of resources reservation, namely active and passive reservation. The agent serving the MN is called a serving proxy, while the agents to which the MN may move from the serving agent are referred to as remote proxy agents. MRSVP functions as follows: the MN sends a Mobility SPECification message (MSPEC) to the CN. This message contains remote proxies IP addresses. The CN sends then active PATH messages to the serving as well as remote proxies. The serving proxy responds by sending an active RESV message, while each remote proxy sends a passive RESV message. Passive RESV messages result in reserving resources passively for the MN. When the MN moves to one of the remote proxies, it activates the resources reserved passively and so on.

HMRSPV integrates RSVP with MIPRR [13]. The RSVP session between the MN and CN is split in the Gateway Foreign Agent (GFA) controlling the MIPRR domain. The MN is assigned two Care of Addresses (CoAs), a Domain CoA (DCoA) and Local CoA (LCoA). The MN registers the DCoA with the Home Agent (HA). Notice that this address represents the GFA, while the LCoA represents the point of attachment and is registered with the GFA. Resources reservation outside the access network is configured for the stable DCoA. When the MN moves inside the domain, it only reestablishes the session to the GFA. To accelerate the handoff and resources reservation when movements between FAs belonging to different MIPRR domains are possible, resources are reserved passively in neighbor FAs not belonging to the current domain where the MN may move to.

Simple QoS integrates between RSVP and MIPv4. This protocol solves the problems related to packets tunneled from the HA to the MN's CoA. This is achieved by establishing an extra RSVP session between the HA and CoA to serve tunneled packets. Fig. 11 presents the handoff procedure and resources reservation on downlink. When the MN moves to a new Foreign Agent (FA), it registers the new CoA with the HA. Once the HA is notified, it establishes RSVP-tunnel between itself and the new FA. Once the CN sends a PATH message to the MN, the PATH message is intercepted by the HA and forwarded through the tunnel towards the MN. Upon receiving the encapsulated PATH message by the FA, it decapsulates and sends the message to the MN, which replies a RESV message following the same route of the PATH message. Resource reservation on uplink is totally different since the MN sends a PATH message towards the CN. The PATH message is intercepted by the first crossover router, which replied a RESV message towards the MN. Alternatively, the MN can use a reverse tunnel from the new FA to the HA.

As mentioned previously, QoMIFA integrates RSVP with MIFA. It extends RSVP through introducing a new object called "Mobility object" used to

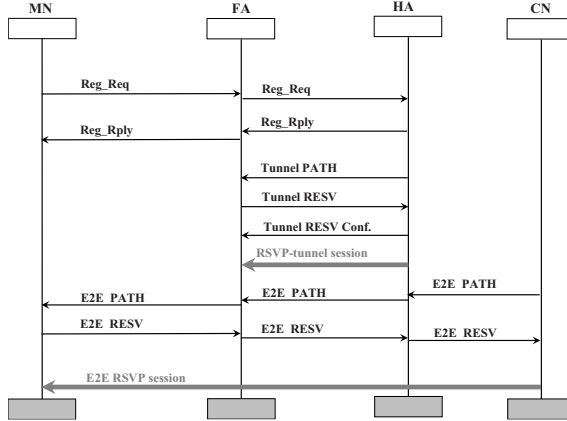


Fig. 1. Handoff procedure of Simple QoS protocol (downlink scenario)

encapsulate control messages of MIFA. The operation of QoMIFA can be briefly described as follows: the current FA serving the MN determines groups of neighbor FAs where the MN may move to from the current FA. Neighbor FAs are notified of the MN in advance of the handoff occurrence. This notification results in storing RSVP path states for the MN in each neighbor FA. Resources, however, will not be reserved passively. Once the MN moves to one of these neighbors, it sends a PATH message, containing the Registration Request(RegRqst) message encapsulated in the “Mobility object”, to the new FA. The new FA in turn exchanges necessary PATH and RESV messages containing MIFA control messages with the old FA. Thereafter, the new FA sends a RESV message to the MN including the Registration Reply(RegRply) encapsulated in the “Mobility object”, see Fig. 2. As a result, the uplink session between the old

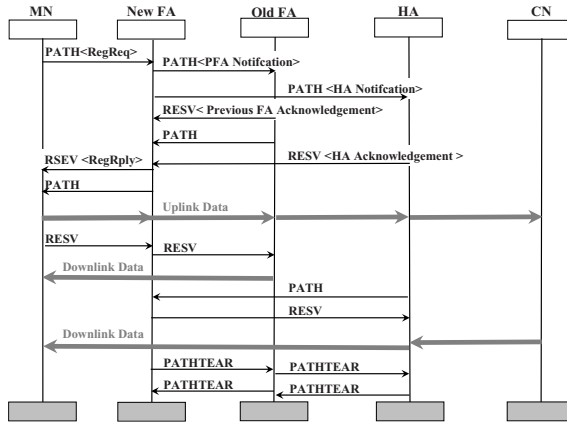


Fig. 2. Handoff procedure of QoMIFA

and new FA is established. Once the MN receives a PATH message from the old FA, it sends a RESV message to establish the downlink session. Thus, a bidirectional session is established between the old FA and MN to forward the MN's data packets until the HA is informed and a new reservation is established. After that, the old resources will be released by means of PATHTEAR message.

The main results can be summarized as follows: hard coupled solutions are more efficient. However, they are more complex and less applicable. The opposite is seen by loose coupled solutions, which are less efficient, less complex and more applicable. The best performance is achieved by hybrid solutions, which inherit the advantages of hard as well as loose coupled solutions. Table. 1 provides a detailed comparison between the approaches coupling between mobility management techniques and QoS mechanisms.

Table 1. A qualitative comparison between the approaches coupling between mobility management techniques and QoS mechanisms

RSVP problems	WLRP	HMRSVP	MRSVP	LRSVP	Simple QoS	QoMIFA
Tunneling problem	Yes	Yes	Yes	Yes	Yes	Yes
Triangular routing problem	Yes	Yes	Yes	Yes	Yes	Yes
Elements supporting QoS	MN, HA, FA, CN, all FAs	MN, HA, FA, CN, GFA	MN, HA, FA, CN, all remote Proxies	MN, AR, the proxy, crossover node	MN, HA, FA, CN	MN, HA, FA, CN
Doubled resources during handoffs	No	No	No	No	No	No
Security	No	No	No	Yes	No	Yes
Avoiding over-reservation in all subnets	No	Yes	No	Yes	Yes	Yes
Route recovery for handoffs	Up to the old FA	Up to the GFA	Up to the anchor node	Up to a crossover node	Up to the HA	Up to the old FA
Passive reservation	Yes	Only for inter domain handoffs	Yes	No	No	No

3 Performance Analysis

So as to evaluate QoMIFA compared to Simple QoS, both are modeled in ns2 and simulated deploying the same network topology under same assumptions. Simple QoS is selected as a candidate to compare with QoMIFA due to the fact that Simple QoS requires minimal changes to the network architecture, thus, it is simple to be employed in existing access networks. Our evaluation comprises studying the impact of network load on the time required to reserve resources, number of dropped packets per handoff and number of packets sent as best-effort until the reservation is accomplished. The following describes the applied simulation scenario and discusses the obtained results.

3.1 Simulation Scenario

The evaluation was achieved deploying a hierarchical network topology as depicted in Fig. 3.

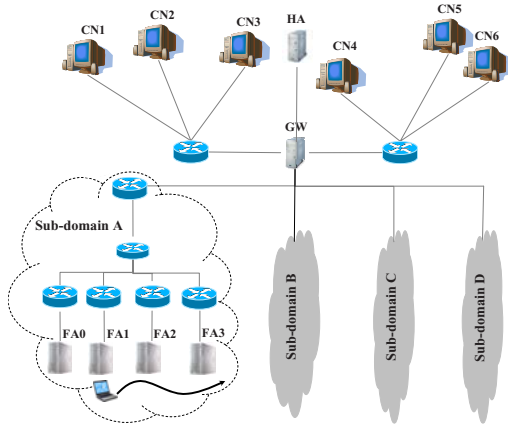


Fig. 3. Used network topology

A domain of 4 sub-domains, each has the same structure, is used. Each sub-domain includes 4 FAs. The distance between cells center of each two neighbor FAs is 198 m. A Gateway (GW) is placed on the uppermost level of the hierarchy in the domain and interconnects the domain with other nodes. The distance between the GW and each FA in the domain is 4 hops. There exist 160 MNs in the domain, 10 equally distributed MNs in the coverage area of each FA. All MNs are registered with the same HA, which is placed outside the domain. Active MNs communicate with 6 CNs placed outside the domain as well. The transmission delay on each link between each two subsequent hops inside the domain is 5 msec. The link between the HA and GW has a transmission delay of 25 msec, while the links between the GW and CN1, CN2, CN3, CN4, CN5 and CN6 have delays of 27, 23, 28, 27, 23 and 28 msec, respectively. All links have a bandwidth of 100 Mbit/s. During the simulation, an active MN is tracked. The selected MN moves from FA0 to FA15 (in sub-domain D) at a speed of 36 km/h.

As mentioned previously, we aim at a detailed analysis of the load impact. For this purpose, 80 MNs are made active, while the remaining 80 MNs stay in idle mode. Active MNs exchange constant bit rate UDP uplink and downlink streams (each has a packet arrival rate of 20 packets per second and a fixed packet size of 500 bytes) with CNs, while idle MNs only produce signaling traffic. The number of active MNs in the range of each FA is changed among 2, 3 and 4 in addition to the observed MN as depicted in Fig. 4. In order to stress the simulation results, several measurements were achieved. More concrete, each scenario was repeated 10 times, which resulted in 150 handoffs for each measurement.

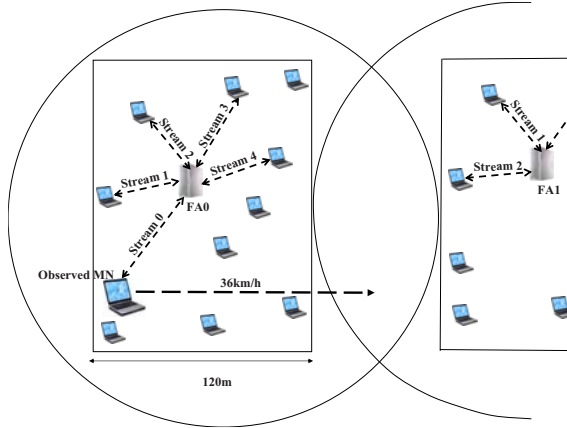


Fig. 4. Load distribution in the range of each FA

3.2 Resources Reservation Latency

Fig. 5 presents the distribution function of the time required to reserve resources on uplink when employing QoMIFA and Simple QoS. The time required to reserve resources on uplink is defined as the time duration required to achieve a handoff and reserve resources for uplink UDP streams.

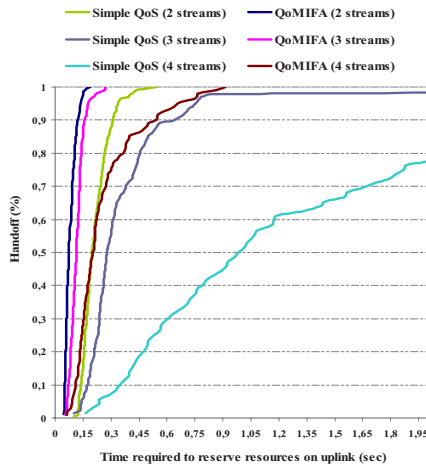


Fig. 5. Distribution function of time required to reserve resources on uplink when employing QoMIFA and Simple QoS under different network loads

This figure shows that the time required to reserve resources on uplink after the handoff employing QoMIFA is significantly reduced compared to Simple

QoS. The reason lies in the coupling between mobility and RSVP control messages in the case of QoMIFA. This coupling results in performing a handoff and simultaneously reserving resources. This is not the case for Simple QoS, which executes the handoff first and reserve resources following that. Let us consider the situation where each FA serves two active MNs, each has a bidirectional background stream on downlink as well as on uplink. While resources on uplink are reserved in less than 187 msec after all handoffs employing QoMIFA, only 43.75 % of the reservations have been accomplished in less than 187 msec employing Simple QoS. QoMIFA remains performing well when increasing the number of active MNs to be 3 in the range of each FA. Reservation of resources on uplink requires less than 187 msec in 96.22 % of the handoffs, while Simple QoS reserves resources on uplink in only 13.14 % of the handoffs in less than 187 msec. Increasing the load in the network so that 4 active MNs are located in the range of each FA, results in a deterioration of the performance of both protocols. The figure shows that QoMIFA requires no more than 187 msec to reserve resources on uplink for 44.55 % of the handoffs. In contrast, clear performance deterioration is seen by Simple QoS. According to the simulation results, the minimum latency required to reserve the required resources on uplink is 162.89 msec.

Similar results are derived from Fig. 6, which shows the minimum, average and maximum latency required to reserve resources on uplink employing the both studied protocols under the mentioned loads.

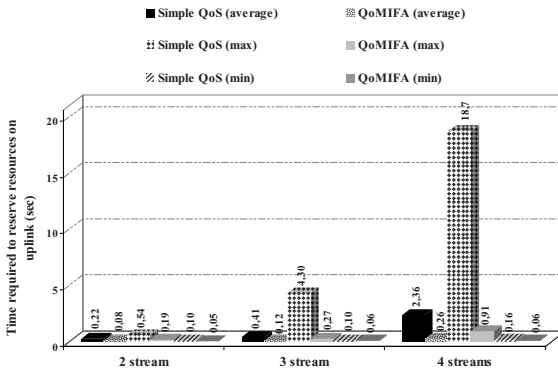


Fig. 6. Minimum, average and maximum time required to reserve resources on uplink when employing QoMIFA and Simple QoS under different network loads

Simulation results show that QoMIFA is 61.64 %, 71.2 % and 88.91 % better than Simple QoS with respect to the average time required to reserve resources on uplink after the handoff when the number of active MNs in the range of each FA varies between 2, 3 and 4, respectively. Notice that the difference between the performance of QoMIFA and Simple QoS increases as the load in the network increases, which means that Simple QoS is more load-sensitive than QoMIFA.

Let us now study the time required to reserve resources for downlink traffic, see Fig. 7 which presents the distribution function of the time required to reserve resources on downlink when employing QoMIFA and Simple QoS in the applied topology under different loads. Again, the time required to reserve resources on downlink is determined as the time duration required to complete the handoff as well as to reserve resources for the downlink UDP streams.

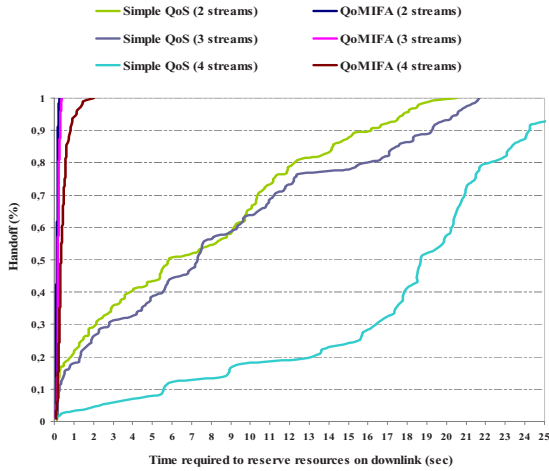


Fig. 7. Distribution function of the time required to reserve resources on downlink when employing QoMIFA and Simple QoS under different network loads

The first result that can be obtained from this figure is that QoMIFA is significantly better than Simple QoS. This is because QoMIFA requires only contacting its old FA to reserve bidirectional RSVP-tunnel between the previous and new CoA of the MN. In contrast, Simple QoS has to contact its HA and establish RSVP-tunnel between the HA and new FA. Another result that can be observed is that the performance of QoMIFA with respect to the time required to reserve resources on downlink is comparable to its performance regarding the time required to reserve resources for the uplink traffic. More concrete, for 2 active MNs in the range of each FA, the resources on downlink are reserved in less than 247 msec after all handoffs employing QoMIFA, while only approximately 14 % of the reservations can be accomplished in less than 247 msec employing Simple QoS. QoMIFA still outperforms Simple QoS when increasing the number of active MNs to be 3 in the range of each FA. For instance, resources reservation on downlink requires less than 247 msec in 81 % of the handoffs employing QoMIFA, while Simple QoS achieves the reservation on downlink in only about 10 % of the handoffs in less than 247 msec. Increasing the load so that each FA serves 4 active MNs affects significantly the performance of Simple QoS. QoMIFA performance is affected as well, it remains, however, better than Simple

QoS. Reserving the resources on downlink requires less than 247 msec in 25 % of the handoffs when employing QoMIFA, while Simple QoS requires less than 247 msec to reserve resources on downlink in only 2 % of the handoffs.

Notice that Simple QoS performs in networks where each FA serves 2 active MNs better than when each FA serves 3 active MNs in approximately 54 % of the handoffs. For networks with 2 and 3 active MNs in the range of each FA, Simple QoS performs comparable in about 10 % of the handoffs. In the remaining handoffs, Simple QoS is better when there are only 2 active MNs in the range of each FA. This is because of the random nature of the simulation.

Simple QoS consumes significantly more time to complete the reservation for the downlink traffic than that required to reserve resources for the uplink traffic. The reason for this is that the PATH message sent towards the HA from the MN operating Simple QoS is intercepted by the crossover router existing on both the path between the HA and old FA and the path between the HA and new FA. The crossover router answers directly by a RESV message. Notice that we do not use a reverse tunnel between the new FA and HA for the uplink traffic, see [12]. On the contrary, the PATH and RESV message are exchanged between the MN and HA for the downlink traffic since the PATH message is initiated by the HA.

Similar results to those derived from Fig. 7 can be observed in from Fig. 8, which presents the minimum, average and maximum time required to reserve resources on downlink employing QoMIFA and Simple QoS under the mentioned loads.

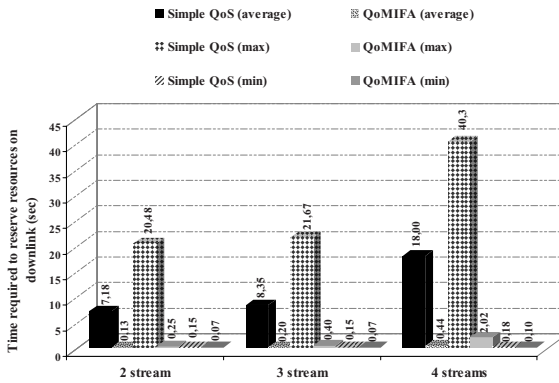


Fig. 8. Minimum, average and maximum time required to reserve resources on downlink when employing QoMIFA and Simple QoS under different network loads

The figure shows a significant performance improvement employing QoMIFA compared to Simple QoS. The reason has been discussed above while discussing Fig. 7. According to the simulation results, QoMIFA outperforms Simple QoS by 98.16 %, 97.56 % and 97.53 % with respect to the average time required to reserve resources on downlink after the handoff when the number of active MNs

in the range of each FA varies between 2, 3 and 4, respectively. Again, this figure shows that QoMIFA is less load-sensitive than Simple QoS. The main reason behind this behavior is that QoMIFA requires only contacting its old FA and reserve resources between this old FA and the new one. Thus, only the load between the old FA and new one is of importance for QoMIFA. For Simple QoS, control messages should climb up to the HA. Therefore, the load in the core network affects strongly the performance of Simple QoS.

3.3 Number of Dropped Packets

Fig. 9 shows the minimum, average and maximum number of dropped packets per handoff on uplink employing QoMIFA and Simple QoS in the used topology under the mentioned loads. The results show that QoMIFA performs 51.26 %, 37.64 % and 50.91 % better than Simple QoS when the number of active MNs in the range of each FA varies between 2, 3 and 4, respectively. This is because QoMIFA performs the registration and reservation of resources simultaneously, while Simple QoS performs the handoff first and reserves resources on uplink after that.

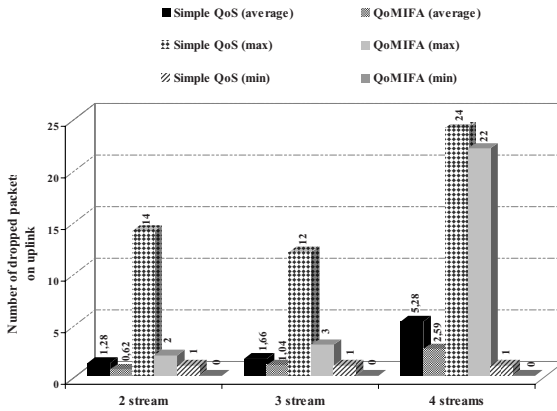


Fig. 9. Minimum, average and maximum number of dropped packets per handoff on uplink resulting from employing QoMIFA and Simple QoS under different network loads

Fig. 10 shows the minimum, average and maximum number of dropped packets per handoff on downlink employing QoMIFA and Simple QoS in the used topology under the mentioned loads. Notice that the average, minimum and maximum number of dropped packets per handoff increase while increasing the load. This is also expected since the handoff latency increases as network load increases, thus, more packets get lost during the handoff. The figure shows also that Simple QoS results in significantly more dropped packets than QoMIFA under all studied loads. The reason behind this behavior is the fast handoffs

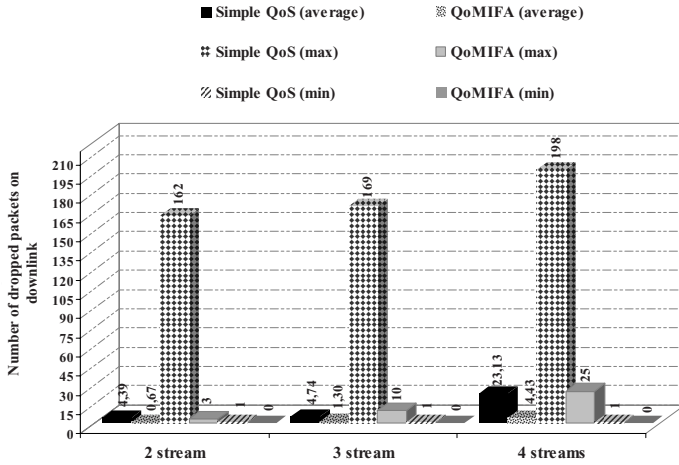


Fig. 10. Minimum, average and maximum number of dropped packets per handoff on downlink resulting from employing QoMIFA and Simple QoS under different network loads

achieved by QoMIFA that only requires, as mentioned previously, contacting its old FA. In contrast, Simple QoS registers with the HA each time the MN moves in the network. The registration with the HA consumes long time especially if the network is high loaded. According to the achieved results, QoMIFA performs 84.71 %, 72.52 % and 80.86 % better than Simple QoS when the number of active MNs in the range of each FA varies between 2, 3 and 4, respectively.

3.4 Number of Best-Effort Packets

This section analyzes the number of packets sent toward the MN as best-effort packets. As known, best-effort packets are the packets sent to the MN after the completion of the handoff and until the resources for downlink traffic are reserved. Fig. 11 shows the minimum, average and maximum number of packets sent on downlink as best-effort employing QoMIFA and Simple QoS in the applied network topology under the mentioned loads.

The figure shows that the number of packets sent as best-effort employing QoMIFA is minimized. This is because of the coupling between mobility support and resources reservation in QoMIFA. The old FA gets informed as it receives the PATH message from the MN via the new FA. It starts, therefore, at this time sending packets as best-effort and simultaneously reserves resources for the downlink traffic. Due to the fast reservation QoMIFA achieves, only few packets are sent as best-effort. In contrast, the MN operating Simple QoS registers first with the HA. After the HA gets informed, it begins forwarding data packets as best-effort to the new CoA of the MN. After the MN completes the handoff, it starts reserving resources for the downlink traffic. This consumes, of course, a considerable time, which in turn results in forwarding considerable amount

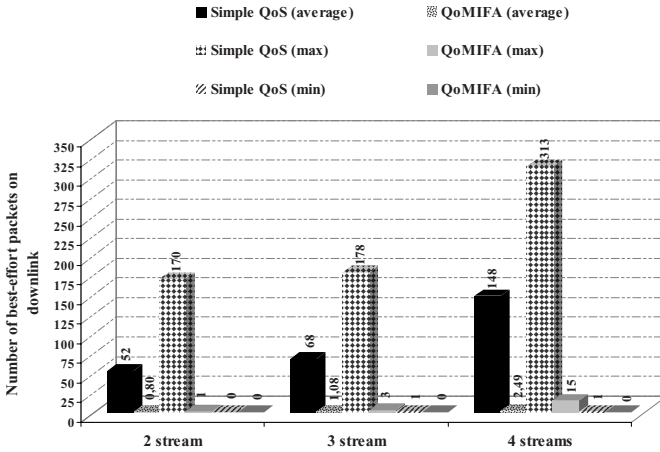


Fig. 11. Minimum, average and maximum number of best-effort packets sent on downlink when employing QoMIFA and Simple QoS under different network loads

of data packets without QoS guarantee. Clearly, this degrades the performance and is not desirable. According to our simulation results, Simple QoS forwards approximately 98.41 % more packets as best-effort than QoMIFA does.

4 Conclusion

In this paper we have evaluated the performance of QoMIFA and Simple QoS under different loads. The evaluation has been achieved by means of simulations for both protocols using ns2. The evaluation comprises the investigation of network load impact on both protocols with respect to the time required to reserve resources, number of dropped packets per handoff and number of downlink packets sent as best-effort after the handoff is completed and until the resources are reserved.

Our simulation results have shown that QoMIFA is capable of achieving fast and smooth handoffs in addition to reserving resources very quickly. This is due to the hybrid coupling between MIFA and RSVP, which enables a simultaneous support of mobility as well as QoS. QoMIFA outperforms Simple QoS under all studied loads. It reserves resources very quickly, minimizes the number of dropped packets per handoff and minimizes the number of packets sent as best-effort. Our results have shown that QoMIFA is approximately 97.75 % and 73.92 % faster than Simple QoS with respect to the average time required to reserve resources on downlink and uplink, respectively. For the average number of dropped packets per handoff on downlink and uplink, QoMIFA is 79.63 % and 46.6 % better, while regarding the number of packets sent as best-effort on downlink, QoMIFA is approximately 98.40 % better.

Currently, we are studying the impact of the network load on TCP throughput. Moreover, we are investigating the impact of MNs speed on the performance of both protocols deploying UDP as well as TCP traffic.

References

1. Perkins, C. (ed.): IP Mobility Support for IPv4. RFC 3344 (August 2002)
2. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775 (June 2004)
3. Diab, A., Mitschele-Thiel, A., Xu, J.: Performance Analysis of the Mobile IP Fast Authentication Protocol. In: 7th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2004), Italy (2004)
4. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation Protocol (RSVP) Version1 Functional Specification. RFC 2205 (September 1997)
5. Alnasouri, E., Mitschele-Thiel, A., Boeringer, R., Diab, A.: QoMIFA: A QoS Enabled Mobility Management Framework in ALL-IP Network. In: 11th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2006), Finland (2006)
6. Network simulator ns-2, <http://www.isi.edu/nsnam/ns/>
7. Galindo-Sánchez, L., Ruiz-Martínez, P.: QoS and Micro mobility Coupling. European Journal for the Informatics Professional, Upgrade (April 2005)
8. Parameswaran, S.: WLRP: A Resource Reservation Protocol for Quality of Service in Next-Generation Wireless Networks. In: 28th Annual IEEE International Conference on Local Computer Networks(LCN 2003) (October 2003)
9. Manner, J., Raatikainen, K.: Localized QoS Management for Multimedia Applications in Wireless Access Networks. University of Helsinki (2004)
10. Talukdar, A.K., et al.: MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts. In: ACM Wireless Networks (January 2001)
11. Tseng, C.-C., et al.: HMRSVP: A Hierarchical Mobile RSVP Protocol. In: International Conference on Distributed Computing Systems Workshop (April 2001)
12. Terzis, A., Srivastava, M., Zhang, L.: A Simple QoS Signaling Protocol for Mobile Hosts in the Integrated Services Internet. In: 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1999), USA (1999)
13. Gustafsson, E., Jonsson, A., Perkins, C.: Mobile IPv4 Regional Registration. RFC 4857 (June 2007)

Resource-Optimized Quality-Assured Ambiguous Context Mediation in Pervasive Environments

Nirmalya Roy¹, Christine Julien¹, and Sajal K. Das²

¹ The Department of Electrical and Computer Engineering
The University of Texas at Austin
{nirmalya.roy, c.julien}@mail.utexas.edu

² The Department of Computer Science and Engineering
The University of Texas at Arlington
das@uta.edu

Abstract. Pervasive computing applications envision sensor rich computing and networking environments that can capture various types of contexts of inhabitants of the environment, such as their locations, activities, vital signs, and environmental measures. Such context information is useful in a variety of applications, for example to manage health information to promote independent living in “aging-in-place” scenarios. In reality, both sensed and interpreted contexts are often ambiguous, leading to potentially dangerous decisions if not properly handled. Thus, a significant challenge facing the development of realistic and deployable context-aware services for pervasive computing applications is the ability to deal with these ambiguous contexts. In this paper, we propose a resource optimized quality assured context mediation framework for resource constrained sensor networks based on efficient context-aware data fusion and information theoretic sensor parameter selection for optimal state estimation. The proposed framework provides a systematic approach based on dynamic Bayesian networks to derive context fragments and deal with context ambiguity or error in a probabilistic manner. Experimental results using SunSPOT sensors demonstrate the promise of this approach.

Keywords: Context-awareness, Ambiguous contexts, Bayesian networks, Multi sensor fusion, Information theory, SunSPOT.

1 Introduction

Recent research in smart environments offers promising solutions to the increasing needs of pervasive computing applications; our work has demonstrated the use of such environments to support the elderly in home based healthcare applications [21]. Essential to such applications is *human-centric* computing and communication, where computers and devices adapt to users’ needs and preferences.

We focus on the computational aspect of user-centric data to provide context-aware services; we demonstrate this through an application for intelligent independent living. Given the expected availability of multiple sensors of different

types, we view context determination as an estimation problem over multiple sensor data streams. Though sensing is becoming increasingly cost-effective and ubiquitous, the interpretation of sensed data as context is still imperfect and ambiguous. Therefore, a critical challenge facing the development of realistic and deployable context-aware services is the ability to handle ambiguous contexts. The conversion of raw data into high-level context information requires processing data collected from heterogeneous distributed sensors through filtering, transformation, and even aggregation, with a goal to minimize the ambiguity of the derived contexts. This context processing could involve simple filtering based on a value match, or sophisticated data correlation, data fusion or information theoretic reasoning techniques. Only with reasonably accurate context(s), can applications be confident to make high quality adaptive decisions. Contexts may also include various aspects of relevant information; they may be instantaneous or durative, ambiguous or unambiguous. Thus, the mapping from sensory output to the context information is non-trivial. We believe context-aware mediation plays a critical role in improving the accuracy of the derived contexts by reducing their ambiguity, although the exact fusion or reasoning technique to use is application and domain specific.

1.1 Related Work

Pervasive computing applications such as the Aware Home [18], Intelligent Room [5] and House_n [13] do not provide explicit reusable support for users to manage uncertainty in the sensed data and its interpretation, and thereby assume that sensed contexts are unambiguous. Toolkits enable the integration of context into applications [8], however, they do not provide mechanisms for sensor fusion or reasoning about contexts' ambiguity. Although other work has proposed mechanisms for reasoning about contexts [25], it does not provide well defined context-aware data fusion models nor address the challenges associated with context ambiguity. Distributed mediation of ambiguous contexts in aware environments [7] has, however, been used to allow the user to correct ambiguity in the sensed input.

Middleware has also effectively supported context-aware applications in the presence of resource constraints (e.g., sensor networks), considering requirements for sensory data or information fusion [1]. DFuse [15] facilitates dynamic transfer of application level information into the network to save power by dynamically determining the cost of using the network. In adaptive middleware for context-aware applications in smart homes [11], the application's quality of context (QoC) requirements are matched with the QoC attributes of the sensors through a utility function. Similarly, in MiLAN [10], applications' quality of service (QoS) requirements are matched with the QoS provided by the sensor network. However, the QoS requirements of the applications and available from the sensors are assumed to be predetermined and known in advance. In pervasive computing environments, the nature (number, types and cost of usage, and benefits) of such sensors available to the applications usually vary, and it is impractical to include a priori knowledge about them. Entropy-based sensor

selection heuristic algorithms [9,16,26] take an information theoretic approach, where the belief state of a tracked object’s location is gradually improved by repeatedly selecting the most informative unused sensor until the required accuracy level of the target state is achieved. The selection of the right sensor with the right information at the right moment was originally introduced in [24], while the structure of an optimal sensor configuration constrained by the wireless channel capacity was investigated in [2]. By eliminating the simplifying assumption that all contexts are certain, we design a context-aware data fusion algorithm to mediate ambiguous context using dynamic Bayesian networks. An approach to intelligent sensor management that provides optimal sensor parameter selection in terms of reduction in ambiguity in the state estimation process has not been considered before. We propose a quality of context function to satisfy the application quality requirements and take an information theoretic approach to decide an optimal sensor configuration.

1.2 Our Contributions

Our approach fuses data from disparate sensors, represents abstract context state, and reasons efficiently about this state, to support context-aware services that handle ambiguity. Our goal is to build a framework that resolves information redundancy and also ensures the conformance to the application’s quality of context (QoC) bound based on an optimal sensor configuration. We state an optimization problem using a generic QoC function to determine the optimal tolerance range of the sensors that satisfy the specified quality of context at a minimum communication cost. Then we propose a Dynamic Bayesian Networks (DBNs) [14] based model that uses the sensed data to interpret context state through fusion and an information theoretic reasoning technique to select the optimal sensor data values to minimize ambiguity. We build a system using various SunSPOT sensors for sensing and mediating user context state. Experiments demonstrate that the proposed framework is capable of determining the user context state and reducing the sensing overhead while ensuring acceptable context accuracy.

This paper is organized as follows. Section 2 describes the basic concepts of our context model and the quality of context (QoC) optimization problem. Section 3 describes the context-aware data fusion model based on DBNs for resolving ambiguity. In Section 4 we study the structure of an optimal sensor configuration to minimize the state estimation error from an information theoretic point of view. We evaluate our approach in Section 5, and Section 6 concludes.

2 Context Model

Context-aware data fusion in the face of ambiguities is challenging because the data in sensor networks is inherently uncertain. We make use of a space-based context model [19] and extend it with quality of context (QoC) attributes. This model captures the underlying description of context related knowledge such as context attribute (a_i), context state (S_i) and situation space (\mathcal{R}_i), and attempts

to incorporate various intuitions that should impact context inference to produce better fusion results as shown in Fig. 1. For specific definitions of these parameters see [22].

2.1 Quality of Context Model

Despite recent developments in sensing and network technology, continuous monitoring of context is still challenging due to resource constraints. Consequently, the amount of information transmitted to a fusion mediator should be minimized to prolong network lifetime. The idea of exploiting temporal correlation across successive samples of individual sensors to reduce communication overhead is addressed in [4]. The focus there was on meeting the quality requirements for a particular class of *aggregation queries*, whereas we focus on arbitrary relationships between a context state and the underlying sensor data. Thus we define Quality of Context (QoC) [12] as a metric for minimizing resource usage. We assume that the application processes an aggregation query with its QoC specified by a precision range Q , which implies that the aggregate value computed at the mediator at any instant should be accurate within $\pm Q$.

We aim to evaluate the update cost of a sensory action A for a given task while ensuring the conformance to the application's QoC bound. Let us denote the update cost (in terms of communication overhead) as \mathcal{U}_i^j if indeed sensor B_i has to report its sample value at time j . Then, we aim to minimize $\sum_{i \in B_m} \mathcal{U}_i(q_i)$, where \mathcal{U}_i denotes the expected average update cost and explicitly indicates its dependence on the specified precision interval q_i (tolerance range). Intuitively, \mathcal{U}_i is inversely proportional to q_i , since the value of the reporting cost increases as the interval shrinks. This update cost also depends on the hop count h_i , the length of the uplink path from sensor B_i to the mediator. Accordingly, minimizing the update cost can be rewritten as: minimize $\sum_{i \in B_m} \mathcal{U}_i(q_i, h_i)$. If the underlying data samples evolve as a random-walk model [12], we have $\mathcal{U}_i \propto \frac{h_i}{(q_i^2)}$ resulting in the following optimization function: $\text{minimize } \sum_{i \in B_m} \frac{h_i}{(q_i^2)}$.

To define the QoC function, we consider three parameters associated with the context attribute: q (the accuracy range of sensor data), Q (the accuracy range of the derived context attribute) and ϕ (the fidelity of the context attribute being derived). Thus, the QoC function is $\phi = f_1(q_1, Q)$ for sensor B_1 . In other words, given tolerances on q_1 and Q , we can say how often (in an ergodic sense), the fused context attribute estimation will lie within $\pm Q$. Similarly, when we consider two sensors B_1 and B_2 jointly, the QoC function should be $\phi = f_{12}(q_1, q_2, Q)$. In this way, for m sensors, there are $2^m - 1$ (all possible combinations except no sensors) functions $f(\cdot)$, indicating the relationship between context attribute, context fidelity, and precision range. Given these continuous functions, the application now says that it needs a precision bound (on the context attribute) of \hat{Q} with a fidelity of at least $\hat{\phi}$. Then, the problem is:

Problem 1. Find the combination of q_1, q_2, \dots, q_m that satisfies $f_{1, \dots, m}(q_1, q_2, \dots, q_m, \hat{Q}) \geq \hat{\phi}$, and yet minimizes $\sum_{i \in B_m} h_i / (q_i)^2$.

The problem of optimally computing the q_i values can be represented by the Lagrangian:

$$\text{minimize } \sum_{i=1}^n \frac{h_i}{q_i^2} + \lambda \times \left[f_{1,\dots,m}(q_1, q_2, \dots, q_m, \dot{Q}) - \phi' \right]. \quad (1)$$

Finding an exact solution to Eqn 1 for any arbitrary $f(\cdot)$ is an NP-complete problem [3], though there are certain forms of $f(\cdot)$ that prove to be more tractable. An attractive case occurs when the i^{th} sensor's individual QoC function has the form $f_S(i) = \nu_i * \exp^{-\frac{q_i^2}{\eta_i}}$, where η_i and ν_i are sensitivity constants for sensor s_i . A larger value of η_i indicates a lower contribution from sensor s_i to the inference of context state S . Moreover, for a selection of m sensors, the resulting $f(\cdot)$ function has the form:

$$f_S(m) = 1 - \prod_{i \in m} (1 - f_S(i)) \quad (2)$$

We solve this by taking the Lagrangian optimization, i.e, we solve for

$$\text{minimize } \sum_{i \in m} \frac{h_i}{q_i^2} + \lambda \left[1 - \prod_{i \in m} [1 - (\nu_i * \exp^{-\frac{q_i^2}{\eta_i}})] - \phi' \right]. \quad (3)$$

and prove the following Lemma.

Lemma 1. *The combination of q_1, q_2, \dots, q_m that satisfies the QoC function $f_{1,\dots,m}(q_1, q_2, \dots, q_m, \dot{Q}) \geq \phi'$ and minimizes the objective function is*

$$\frac{h_1 * \eta_1 * (1 - \nu_1 * \exp(-\frac{q_1^2}{\eta_1}))}{q_1^4 * \nu_1 * \exp(-\frac{q_1^2}{\eta_1})} = \dots = \frac{h_m * \eta_m * (1 - \nu_m * \exp(-\frac{q_m^2}{\eta_m}))}{q_m^4 * \nu_m * \exp(-\frac{q_m^2}{\eta_m})}$$

Proof. The above expression follows immediately by taking partial derivatives of the Lagrangian in Eqn 3 and setting them to 0 as shown below. In our case:

$$\text{minimize } \sum_{i \in B_m} \frac{h_i}{q_i^2} \quad \text{subject to: } 1 - \prod_{i \in B_m} [1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}}] \geq \phi' \quad (4)$$

Taking log we can rearrange the constraint of Eqn 4

$$\log(1 - \phi') \geq \sum_{i \in B_m} \log(1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}}) \quad (5)$$

Considering this, we form the Lagrangian constraint,

$$\text{minimize } \sum_{i \in B_m} \frac{h_i}{q_i^2} + \lambda \left[\log(1 - \phi') - \sum_{i \in B_m} \log(1 - \nu_i * \exp^{-\frac{q_i^2}{\eta_i}}) \right] \quad (6)$$

Taking the partial derivative of the Eqn 6 with respect to q_i and equating it to 0, we find

$$\lambda = \frac{h_i * \eta_i * (1 - \nu_i * \exp(-\frac{q_i^2}{\eta_i}))}{q_i^4 * \nu_i * \exp(-\frac{q_i^2}{\eta_i})} \quad (7)$$

which proves the optimal choices of q_i from Lemma 11

This optimization problem helps us to choose the values of q_1, q_2, \dots, q_m for a given set of sensors m , that minimizes the total cost while ensuring the required accuracy.

3 Context-Aware Data Fusion

A characteristic of pervasive computing is that applications sense and react to *context*, information sensed about the environment and its occupants, by providing context-aware services that facilitate applications' actions. Here we develop an approach for sensor data fusion in a context-aware environment considering the underlying space-based context model and a set of intuitions it covers; we use a context-aware healthcare example to explicate our model. We propose a DBN based model in our previous work [20] that we briefly outline in the remainder of this section.

3.1 Dynamic Bayesian Network Based Model

Our motivation is to use the data fusion algorithm to develop a context-aware model to gather knowledge from sensor data. Dynamic Bayesian Networks (DBNs) provide a coherent and unified hierarchical probabilistic framework for sensory data representation, integration and inference. Fig. 11 illustrates a DBN based framework for a context-aware data fusion system consisting of a situation space, context states, context attributes, a sensor fusion mediator and a network of information sensors.

Let us assume a situation space \mathcal{R}_i to confirm using the sensory information sources $B = \{B_1, \dots, B_m\}$, a set of measurements taken from sensors labeled from 1 to m . The context attribute most relevant should decrease the ambiguity of the situation space a_j^R the most; we will select the one that can direct the probabilities of the situation space to near one (for maximum) and zero (for minimum). Let \mathcal{V}_i be the ambiguity reducing utility to the situation space \mathcal{R}_i . Then the expected value of \mathcal{V}_i , given a context attribute a_i^t from sensor B_i , which has K possible values, can be represented as:

$$\mathcal{V}_i = \max_{i=0}^K \sum_{j=0}^N [P(a_j^R | a_i^t)]^2 - \min_{i=0}^K \sum_{j=0}^N [P(a_j^R | a_i^t)]^2 \quad (8)$$

where $i \in \{1, 2, \dots, m\}$ identifies the sensor that provides the attribute. This context attribute can be measured by propagating the possible outcome of an information source, i.e., $P(a_j^R | a_i^t) = \frac{P(a_j^R, a_i^t)}{P(a_i^t)}$.

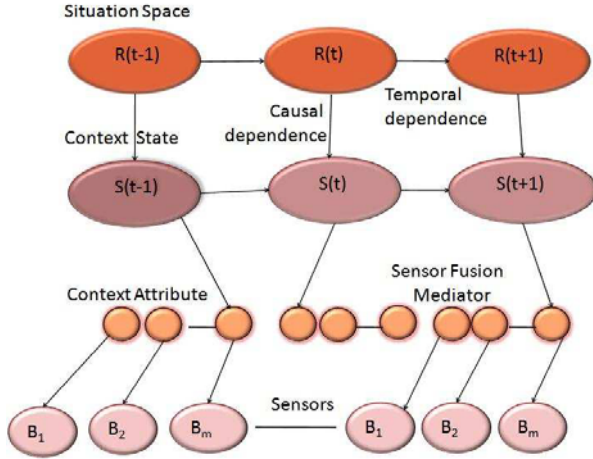


Fig. 1. Context-Aware Data Fusion Framework based on Dynamic Bayesian Networks

Considering the information update cost and ambiguity reducing utility, the overall utility can be expressed as:

$$U_i = \alpha \mathcal{V}_i + (1 - \alpha)(1 - \mathcal{U}_i) \tag{9}$$

where \mathcal{U}_i is the update cost to acquire the information by sensor i with a knowledge of the QoC bound, and α denotes the balance between ambiguity reduction and cost. Eqn. 9 represents the contributions to ambiguity reduction and cost to achieve the desired level of confidence. We can observe from Eqn. 9 that the utility value of a_i increases with the ambiguity reducing utility and decreases with increasing acquisition cost. The most economically efficient disambiguation sensor action A^* can be chosen with the help of the following decision rule: $A^* = \arg \max_A \sum_j U(B, a_j^R) P(a_j^R | B)$; where $B = \{B_1, \dots, B_m\}$ is a set of measurements taken from the sensors labeled from 1 to m at a particular point of time. By incorporating the temporal dependence between the nodes as shown in Fig. 1, the probability distribution of the situation space we want to achieve can be described as: $P(\mathcal{R}, A) = \prod_{t=1}^{T-1} P(S_t | S_{t-1}) \prod_{t=1}^{T-1} P(\mathcal{R}_t | B_t) P(\mathcal{R}_0)$; where T is the time boundary. This sensor action strategy must be recalculated at each time slice since the best action varies with time.

4 Optimal Sensor Parameter Selection

Considering that most sensors are battery operated and use wireless communication, energy-efficiency is important in addition to managing changing QoC requirements. For example, higher quality might be required for certain health-related context attributes during high stress situations such as a medical emergency, and lower quality during low stress situations such as sleep. Fig. 2 shows

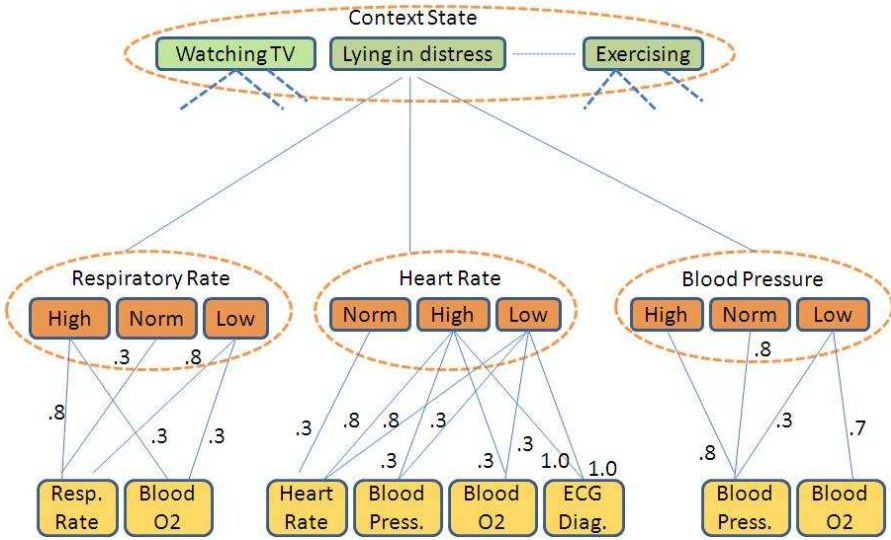


Fig. 2. State-based Context attribute requirement graph with the required QoC

the context attributes requirement graph for a personal health monitor and includes multiple states for each vital signs that can be monitored depending upon the context state of the patient. For example, the Fig. 2 shows that when a patient is lying in a distressed state and the blood pressure is low, the blood oxygen level must be monitored with a quality of .7 and the blood pressure must be monitored with a quality of .8. So the problem here is to decide what type of information each sensor should send to the fusion center to estimate the best current state of the patient while satisfying the application QoC requirements for each context attribute by minimizing the state estimation error.

In this section, we introduce a formalism for optimal sensor parameter selection for state estimation. We define optimality in terms of reduction in ambiguity in the context estimation. The main assumption is that state estimation becomes more reliable and accurate if the ambiguity or error in the underlying state estimation process can be minimized. We investigate this from an information theoretic perspective [6] where information about the context attribute is made available to the fusion center by a set of smart sensors. The fusion center produces an estimate of the state of the situation based on intelligent analysis on the received data. We assume that the noisy observations across sensors are independent and identically distributed (i.i.d) random variables conditioned on the binary situation \mathcal{R} (we assume situation \mathcal{R} here as binary for ease of modeling). Each sensor attribute has a source entropy rate $H(a_i)$. Any sensor wishing to report this attribute must send $H(a_i)$ bits per unit time, which is the entropy of the source being measured assuming that the sensor is sending the exact physical state. Of course, different sensors contribute in different measures to the error in state estimation. So, the problem is to minimize the ambiguity

(or keep it within a specified bound), while not exceeding the shared link rate \mathcal{Q} . Thus by maximizing the a posteriori detector probability we can minimize the estimation error of the random variables based on noisy observations from a set of sensors at the fusion center to accurately reconstruct the state of the situation [2].

Problem 2. *Let B be the vector of sensors and A be the set of attributes, then imagine a $(B \times A)$ matrix where $B_{mi} = 1$ where sensor m sends attribute a_i . Then, the goal is to find a matrix $(B \times A)$ within the capacity constraint \mathcal{Q} which minimizes the estimation error of the situation space.*

$$\sum_m \sum_i H(a_i) * B_{mi} < \mathcal{Q} \quad \text{and} \quad \text{minimize } [P_e = P\{\tilde{\mathcal{R}} \neq \mathcal{R}\}] \quad (10)$$

where $\tilde{\mathcal{R}}$ is an estimate of the original state \mathcal{R} .

4.1 Problem Explanation

We assume \mathcal{R} to be a random variable drawn from the binary alphabet $\{\mathcal{R}_0, \mathcal{R}_1\}$ with prior probabilities p_0 and p_1 , respectively. In our case, each sensor needs to determine a sequence of context attributes for a sequence of context states $\{S_{m,t} : \forall t = 1, 2, \dots, T\}$ about the value of situation \mathcal{R} . We assume that random variables $S_{m,t}$ are i.i.d., given \mathcal{R} , with conditional distribution $p_{S|\mathcal{R}}(\cdot|\mathcal{R}_i)$. The sensors could construct and send a summary $Z_{m,t} = \pi_m(S_{m,t})$ of their own observations to a fusion center at discrete time t . The fusion center then produces an estimate $\tilde{\mathcal{R}}$ of the original situation \mathcal{R} . Thus we need to find an admissible strategy for an optimal sensor-attribute mapping matrix $(B \times A)$ that minimizes the probability of estimation error $P_e = P\{\tilde{\mathcal{R}} \neq \mathcal{R}\}$.

Definition 1. *A set of decision rules π_m for an observation $X \rightarrow \{1, 2, \dots, \bar{a}_m\}$ where \bar{a}_m is the number of attributes admissible to sensor B_m with the admissible strategy denoted by π , consists of an integer M in $(B \times A)$ matrix, such that*

$$\sum_{m=1}^M \sum_i H(\bar{a}_m \cdot a_i) * B_{mi} < \mathcal{Q}$$

The evaluation of message $z_{m,t} = \pi_m(s_{m,t})$ by sensor B_m is forwarded to the fusion center at time t . Since we are interested in a continuous monitoring scheme here, we consider that the observation interval T tends to ∞ . But the associated probability of error at the fusion center goes to zero exponentially fast as T grows unbounded. Thus we can compare the transmission scheme through the error exponent measure or Chernoff information:

$$E(\pi) = - \lim_{T \rightarrow \infty} \frac{1}{T} \log P_e^{(T)}(\pi) \quad (11)$$

where $P_e^{(T)}(\pi)$ denotes the probability of error at the fusion center for strategy π considering the maximum a posteriori detector probability. We use $\Pi(\mathcal{Q})$ to

capture all admissible strategies corresponding to an independent frequently varying multiple access channel with capacity \mathcal{Q} and redefine our problem as follows:

Problem 3. Find an admissible strategy $\pi \in \Pi(\mathcal{Q})$ that maximizes the Chernoff information:

$$E(\pi) = - \lim_{T \rightarrow \infty} \frac{1}{T} \log P_e^{(T)}(\pi) \tag{12}$$

4.2 Results

Let us consider an arbitrary admissible strategy $\pi = (\pi_1, \pi_2, \dots, \pi_M)$ and denote the space of received information corresponding to this strategy by:

$$\gamma = \{1, 2, \dots, \bar{a}_1\} \times \{1, 2, \dots, \bar{a}_2\} \times \dots \times \{1, 2, \dots, \bar{a}_M\} \tag{13}$$

where $(\pi_1(x_1), \pi_2(x_2), \dots, \pi_M(x_M)) \in \gamma$; for all observation vectors $(x_1, x_2, \dots, x_M) \in X^M$. Since the maximization of the a posteriori detector is basically the minimization of the probability of estimation error at the fusion center, we could just approximate this probability of error for a finite observation interval T and measure the error exponent corresponding to strategy π using Chernoff's theorem [6].

Next we consider $p_{\tilde{z}|\mathcal{R}_0}(\cdot|\mathcal{R}_0)$ and $p_{\tilde{z}|\mathcal{R}_1}(\cdot|\mathcal{R}_1)$ as the conditional probability mass functions on γ , given situations \mathcal{R}_0 and \mathcal{R}_1 . Now for $\tilde{z} = (z_1, z_2, \dots, z_M)$ and $i \in 0, 1$:

$$\begin{aligned} p_{\tilde{z}|\mathcal{R}_i}(\tilde{z}|\mathcal{R}_i) &= P_i \{ \tilde{x} : (\pi_1(x_1), \pi_2(x_2), \dots, \pi_M(x_M)) = \tilde{z} \} \\ &= \prod_{m=1}^M P_i \{ \pi_m(u_m) \} \end{aligned} \tag{14}$$

where the probability of event W is $P_i\{W\}$ under situation \mathcal{R}_i , and $\pi_m(u_m) = \{x : \pi_m(x) = z_m\}$.

Theorem 1. Using Chernoff's theorem [6], the best achievable exponent in the probability of error at the fusion center is given by

$$E(\pi) = - \min_{0 \leq k \leq 1} \log \left[\sum_{\tilde{z} \in \gamma} (p_{\tilde{z}|\mathcal{R}_0}(\tilde{z}|\mathcal{R}_0))^k (p_{\tilde{z}|\mathcal{R}_1}(\tilde{z}|\mathcal{R}_1))^{1-k} \right]$$

where $\pi \in \Pi(\mathcal{Q})$ is given. Using Theorem [1] we can restate our original problem as follows

Problem 4. Maximize the Chernoff information

$$E(\pi) = - \min_{0 \leq k \leq 1} \log \left[\sum_{\tilde{z} \in \gamma} (p_{\tilde{z}|\mathcal{R}_0}(\tilde{z}|\mathcal{R}_0))^k (p_{\tilde{z}|\mathcal{R}_1}(\tilde{z}|\mathcal{R}_1))^{1-k} \right]$$

corresponding to an admissible strategy $\pi \in \Pi(\mathcal{Q})$.

The problem of finding the optimal decision rules $\pi = (\pi_1, \pi_2, \dots, \pi_M)$ is hard even when the assignment vector $(\bar{a}_1, \bar{a}_2, \dots, \bar{a}_M)$ is fixed a priori. Hence we try to derive a set of simplified conditions for Problem 4. Thus we state the following Lemma, where we obtain an upper bound of the contribution of a single sensor to the Chernoff information and find sufficient conditions for which having \mathcal{Q} sensors in the $(B \times A)$ matrix, each sending one bit of information, is optimal.

Lemma 2. *For strategy π , the contribution $E_{B_m}(\pi)$ from a single sensor B_m to the Chernoff information $E(\pi)$ is bounded above by the Chernoff information E^* contained in one context state S ,*

$$E_{B_m}(\pi) \leq E^* \equiv -\min_{0 \leq k \leq 1} \log \left[\int_X (p_{S|\mathcal{R}}(x|\mathcal{R}_0))^k \cdot (p_{S|\mathcal{R}}(x|\mathcal{R}_1))^{1-k} dx \right] \quad (15)$$

Proof. Proof shown in the Appendix.

Let us represent $E_1(\pi_m)$ as the Chernoff information corresponding to a single sensor with decision rule π_m , i.e.,

$$E_1(\pi_m) = -\min_{0 \leq k \leq 1} \log \left[\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^k (P_1\{\pi_m(u_m)\})^{1-k} \right] \quad (16)$$

and let Π_b be the set of binary functions on the observation space X .

Lemma 3. *Consider a binary function $\tilde{\pi}_b \in \Pi_b$ such that $E_1(\tilde{\pi}_b) \geq \frac{E^*}{2}$. Then having \mathcal{Q} identical sensors, each sending one bit of information is optimal.*

Proof. Let strategy $\pi = (\pi_1, \pi_2, \dots, \pi_M) \in \Pi(\mathcal{Q})$ and rate \mathcal{Q} be given. We construct an admissible strategy $\pi' \in \Pi(\mathcal{Q})$ such that $E(\pi') \geq E(\pi)$. We divide the collection of decision rules $\{\pi_1, \pi_2, \dots, \pi_M\}$ into two sets; the first set contains all of the binary functions, whereas the other is composed of the remaining decision rules. We also consider I_b to be the set of integers for which the function π_m is a binary decision rule: $I_b = \{m : 1 \geq m \geq M, \pi_m \in \Pi_b\}$. Similarly, we define $I_{nb} = \{1, 2, \dots, M\} - I_b$. Considering the binary decision rule $\hat{\pi}_b \in \Pi_b$, we express $E_1(\hat{\pi}_b) \geq \max\{\max_{m \in I_b} \{E_1(\hat{\pi}_b)\}, \frac{E^*}{2}\}$. Since by assumption $\tilde{\pi}_b \in \Pi_b$ and $E_1(\tilde{\pi}_b) \geq \frac{E^*}{2}$, we infer that such a function $\hat{\pi}_b$ always exists. Observing that $m \in I_{nb}$ implies that $\bar{a}_m \geq 2$, which in turn yields $H(\bar{a}_m \cdot a_i) \geq 2$. Considering the alternative scheme π' , where π' is an admissible strategy, we replace every sensor with index in I_{nb} by two binary sensors with decision rule $\hat{\pi}_b$. This new scheme outperforms the original strategy π as shown in Eqn 17.

$$\begin{aligned} E(\pi') &= (|I_b| + 2|I_{nb}|) E_1(\hat{\pi}_b) \geq |I_b| E_1(\hat{\pi}_b) + |I_{nb}| E^* \\ &\geq \sum_{m=1}^M \left[-\min_{0 \leq k \leq 1} \log \left[\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^k (P_1\{\pi_m(u_m)\})^{1-k} \right] \right] \\ &\geq -\min_{0 \leq k \leq 1} \log \left[\sum_{\bar{z} \in \gamma} \left(\prod_{m=1}^M (P_0\{\pi_m(u_m)\})^k (P_1\{\pi_m(u_m)\})^{1-k} \right) \right] \\ &= E(\pi) \end{aligned} \quad (17)$$

The Chernoff information at the fusion center is monotonically increasing in the number of sensors for a fixed decision rule $\tilde{\pi}_b$. State estimation error can be minimized by augmenting the number of sensors in π' until the capacity constraint \mathcal{Q} is met.

The strategy π being arbitrary, we conclude that having \mathcal{Q} identical sensors in the $(B \times A)$ matrix, each sending one bit of information is optimal in terms of reducing the state estimation error. This configuration also conveys that the gain offered through multiple sensor fusion exceeds the benefits of getting detailed information from each individual sensor.

5 Experimental Components and Evaluation

We use the SunSPOT [23] (Sun Small Programmable Object Technology) device for context sensing and mediation, which is a small, wireless, battery powered experimental platform. Each free-range SunSPOT contains a processor, radio, sensor board and battery; the base-station Sun SPOT contains a processor and radio only. The SunSPOT uses a 32-bit ARM9 microprocessor running the Squawk VM and programmed in Java, supporting the IEEE 802.15.4 standard. In our context sensing and performance evaluation we will use various built-in sensors available with the SunSPOT sensor board.

5.1 Empirical Determination of Context Estimates

We used the accelerometer to measure the tilt value of the SunSPOT (in degrees) when the monitored individual was in three different context states: *sitting*, *walking* and *running*. From the collected samples, we computed the 5th and 95th percentile of the tilt readings, corresponding to each state. Table 1 shows the resulting ranges in the accelerometer tilt readings observed for each of the three states. The results indicate that there is an observable separation in the ranges of the tilt values for the three different states. This suggests that the states can be distinguished reasonably accurately even under moderate uncertainty in the sensor's readings.

Similarly, we also used the SunSPOT light sensor to measure the light level for different user contexts. Intuitively, low values of ambient light intensity may be indicative of a *'sleeping'* state, while higher values of light intensity are likely to result when the individual is *'active'*. Table 2 shows the observed ranges for the light values for each of these two states. The accuracy of context from the light sensor is, however, much lower, as users may often be inactive (e.g., sitting), even under high illumination.

5.2 Measurement of QoC Accuracy and Sensor Overheads

To study the potential impact of varying the tolerance range on each sensor and the resulting tradeoff between the sensor reporting overhead, we collected traces for the SunSPOT motion and light sensors for a single user who engaged

Table 1. Calibrated Accelerometer Sample Values for different Context State

Range(5 – 95th percentile) of Tilt Values (in degree)	Context State
85.21 to 83.33	<i>Sitting</i>
68.40 to 33.09	<i>Walking</i>
28.00 to –15.60	<i>Running</i>

Table 2. Light Sensor Values (lumen) for different Context State

Avg. Range of Light level (lumen)	Context State
LightSensor.getValue() = 10 to 50	Turned on → active
LightSensor.getValue() = 0 to 1	Turned off → sleeping

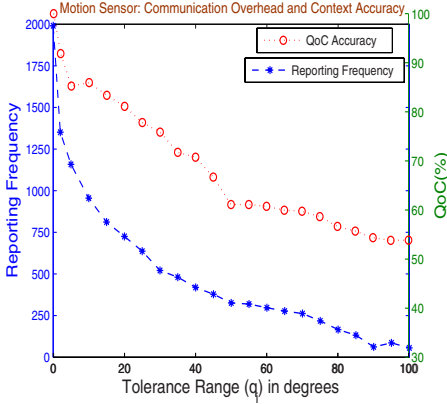


Fig. 3. Communication Overhead & QoC Accuracy vs. Tolerance Range using Motion Sensor

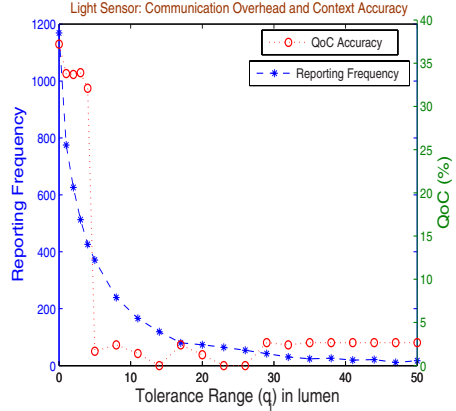


Fig. 4. Communication Overhead & QoC Accuracy vs. Tolerance Range using Light Sensor

in a mix of three different activities (*sitting*, *walking* and *running*) for a total of ≈ 6 minutes (2000 samples at $5.5Hz$). We then used an emulator to mimic the samples that a sensor would have reported, given the trace, for a given q , and compared the context inferred from the values reported by the emulation against the ground truth. Fig. 3 shows the resulting plots for the ‘total number of samples reported’ (an indicator of the reporting overhead) and the corresponding QoC (defined as $1 - error\ rate$) achieved, for different values of the tolerance range (q_m) for the motion sensor. Fig. 4 plots the corresponding values vs. the tolerance range (q_l) for the light sensor.

As the figures demonstrate, there is, in general, a continuous drop in the reporting overhead and the QoC accuracy as q increases. However, as seen in Fig. 3, a QoC of $\approx 80\%$ is achieved for a modestly large q value of 40. Moreover, using this tolerance range reduces the reporting overhead dramatically by $\approx 85\%$ (from 1953 \rightarrow 248). This suggests that it is indeed possible to achieve significant savings in bandwidth, if one is willing to tolerate marginal degradation in the accuracy of the sensed context. A similar behavior is observed for the light sensor ($q = 4$ incurs a 5% loss in QoC vs. a $\approx 65\%$ reduction in reporting overhead). However, as the difference between the lumen ranges for *Active* vs. *Sleeping* is only ≈ 10 (Table 2), increasing q actually leads to a sharp fall in the QoC.

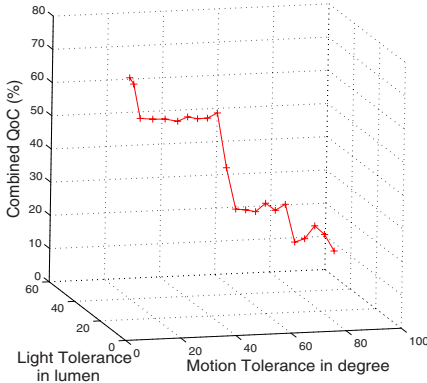


Fig. 5. QoC Accuracy vs. Tolerance Range using both Motion and Light Sensor Together

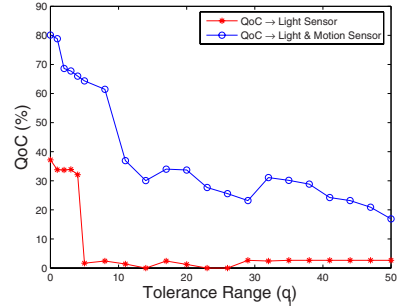


Fig. 6. Comparison of QoC Accuracy Improvement using Multiple Sensor

5.3 The Benefit of Joint Sensing

We also investigated how the use of readings jointly from both sensors affects the inferring accuracy vs. tolerance ranges. We consider the individual to be in a *sitting*, *walking* or *running* state whenever the motion sensor tilt values lie within the corresponding range and the light sensor values indicate an *active* state. Fig. 5 uses a three-dimensional plot to illustrate the observed inferring fidelity when the tuple (q_m, q_l) is jointly varied. This confirms the QoC accuracy is now less susceptible to individual q variations. Fig. 6 confirms this benefit by plotting the QoC vs. q obtained using the light sensor against that obtained by using both light and motion sensors (the q ranges of both being identical). Clearly, the QoC obtainable from the combination of the two sensors is much higher than that of a single sensor. This confirms that the gain obtained by having more sensors exceeds the benefits of getting detailed information from each individual sensor in accordance to our information theoretic analysis. Through this evaluation we observed it is indeed possible to significantly reduce the sensors' resource usage while satisfying the application quality requirements in pervasive care environments.

6 Conclusion

This paper presents a framework that supports ambiguous context mediation based on dynamic Bayesian networks and information theoretic reasoning, exemplifying the approach through context-aware healthcare applications in smart environments. Our framework satisfies the applications' quality requirements based on a resource optimized QoC function, provides a Bayesian approach to

fuse context fragments and deal with context ambiguity in a probabilistic manner, and depicts an information theoretic approach to minimize the error in the state estimation process. A SunSPOT context sensing system is developed and subsequent experimental evaluation is done.

References

1. Alex, H., Kumar, M., Shirazi, B.: MidFusion: An adaptive middleware for information fusion in sensor network applications. Elsevier Journal of Information Fusion (2005)
2. Chamberland, J., Verravalli, V.: Decentralized detection in sensor networks. IEEE Transactions on Signal Processing 51(2), 10 (2003)
3. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J.M., Hong, W.: Model-based Approximate Querying in Sensor Networks. Int'l Journal on Very Large Data Bases, VLDB Journal (2005)
4. Deshpande, A., Guestrin, C., Madden, S.: Using Probabilistic Models for Data Management in Acquisitional Environments. In: Proc. of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR), January 2005, pp. 317–328 (2005)
5. Coen, M.: The future of human-computer interaction or how I learned to stop worrying and love my intelligent room. IEEE Intelligent Systems 14(2), 8–10 (1999)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
7. Dey, A.K., Mankoff, J., Abowd, G.D., Carter, S.: Distributed Mediation of Ambiguous Context in Aware Environments. In: Proc. of the 15th Annual Symposium on User Interface Software and Technology (UIST 2002), October 2002, pp. 121–130 (2002)
8. Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction (HCI) Journal 16(2-4), 97–166 (2001)
9. Ertin, E., Fisher, J., Potter, L.: Maximum mutual information principle for dynamic sensor query problems. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 405–416. Springer, Heidelberg (2003)
10. Heinzelman, W., Murphy, A.L., Carvalho, H.S., Perillo, M.A.: Middleware to Support Sensor Network Applications. IEEE Network 18, 6–14 (2004)
11. Huebscher, M.C., McCann, J.A.: Adaptive Middleware for Context-aware Applications in Smart Homes. In: Proc. of the 2nd Workshop on Middleware for Pervasive and Ad-hoc Computing, October 2004, pp. 111–116 (2004)
12. Hu, W., Misra, A., Shorey, R.: CAPS: Energy-Efficient Processing of Continuous Aggregate Queries in Sensor Networks. In: Fourth IEEE Int'l Conference on Pervasive Computing and Communications (PerCom), pp. 190–199 (2006)
13. Intille, S.S.: The goal: smart people, not smart homes. In: Proc. of the Int'l Conference on Smart Homes and Health Telematics. IOS Press, Amsterdam (2006)
14. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, New York (2001)
15. Kumar, R., Wolenetz, M., Agarwalla, B., Shin, J., Hutto, P., Paul, A., Ramachandran, U.: DFuse: a Framework for Distributed Data Fusion. In: Proc. of the 1st Int'l Conference on Embedded Networked Sensor Systems, November 2003, pp. 114–125 (2003)

16. Liu, J., Reich, J., Zhao, F.: Collaborative in-network processing for target tracking. EURASIP JASP: Special Issues on Sensor Networks 2003(4), 378–391 (2003)
17. Netica Bayesian Network Software, <http://www.norsys.com>
18. Orr, R.J., Abowd, G.D.: The Smart Floor: A Mechanism for Natural User Identification and Tracking. In: Proc. of 2000 Conference on Human Factors in Computing Systems (CHI 2000). ACM Press, New York (2000)
19. Padovitz, S., Loke, W., Zaslavsky, A., Bartolini, C., Burg, B.: An approach to Data Fusion for Context Awareness. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 353–367. Springer, Heidelberg (2005)
20. Roy, N., Pallapa, G., Das, S.K.: A Middleware Framework for Ambiguous Context Mediation in Smart Healthcare Application. In: Proc. of IEEE Int'l Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob) (October 2007)
21. Roy, N., Roy, A., Das, S.K.: Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-learning based Approach. In: Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom), March 2006, pp. 148–158 (2006)
22. Roy, N., Julien, C., Das, S.K.: Resource-Optimized Ambiguous Context Mediation for Smart Healthcare. Technical Report TR-UTEDGE-2008-011, UT-Austin (2008)
23. SunSpotWorld - Home of Project Sun SPOT, <http://www.sunspotworld.com/>
24. Tenney, R.R., Sandell Jr., N.R.: Detection with distributed sensors. IEEE Trans. Aerosp. Electron. Syst. AES-17, 501–510 (1981)
25. Vurgun, S., Philpose, M., Pavel, M.: A Statistical Reasoning System for Medication Prompting. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 1–18. Springer, Heidelberg (2007)
26. Wang, H., Yao, K., Pottie, G., Estrin, D.: Entropy-based sensor selection heuristic for target localization. In: Proc. of IPSN 2004 (April 2004)

Appendix

Proof of Lemma 2: We consider the contribution of sensor B_m . The Chernoff information for strategy $\pi = (\pi_1, \pi_2, \dots, \pi_M)$ is given by

$$\begin{aligned}
 E(\pi) &= - \min_{0 \leq k \leq 1} \log \left[\sum_{\tilde{z} \in \gamma} (p_{\tilde{z}|\mathcal{R}}(\tilde{z}|\mathcal{R}_0))^k (p_{\tilde{z}|\mathcal{R}}(\tilde{z}|\mathcal{R}_1))^{1-k} \right] \\
 &= - \log \left[\prod_{m=1}^M \left(\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^{k^*} (P_1\{\pi_m(u_m)\})^{1-k^*} \right) \right] \\
 &= - \sum_{m=1}^M \log \left[\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^{k^*} (P_1\{\pi_m(u_m)\})^{1-k^*} \right] \\
 &= - \log \left[\sum_{z_1=1}^{\bar{a}_1} (P_0\{\pi_m(u_m)\})^{k^*} (P_1\{\pi_m(u_m)\})^{1-k^*} \right] \\
 &\quad - \sum_{m=2}^M \log \left[\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^{k^*} (P_1\{\pi_m(u_m)\})^{1-k^*} \right] \tag{18}
 \end{aligned}$$

where the Chernoff information $E(\pi)$ is maximized at k^* . So we can conclude that contribution of sensor B_m to the Chernoff information $E(\pi)$ can not exceed

$$- \min_{0 \leq k \leq 1} \log \left[\sum_{z_m=1}^{\bar{a}_m} (P_0\{\pi_m(u_m)\})^k (P_1\{\pi_m(u_m)\})^{1-k} \right] \quad (19)$$

which in turn is upper bounded by the Chernoff information contained in one context state S . So, the Lemma 2 confirms that the contribution of a single sensor to the total Chernoff information can not exceed the information contained in each observation. Hence we derive the sufficient condition based on the Lemma 2 for which having Q binary sensors is optimal.

QShine 2009

Session V – Switches, Systems and the Internet

Fluctuations and Lasting Trends of QoS on Intercontinental Links

Tomasz Bilski

Poznan University of Technology,
Pl. Sklodowskiej-Curie 5, 60-965 Poznan, Poland
tomasz.bilski@put.poznan.pl

Abstract. The paper presents an analysis of short- and long-term changes in the QoS of intercontinental connections. First we will show that despite fast and numerous advances in physical layer, link layer, router capacity and new telecommunication cables deployment, QoS measures are hardly progressing in long-term (years) perspective. Transatlantic (North America – Europe) connections will be thoroughly analyzed. Next we will show that due to submarine cable breakages temporary network performance is unpredictable. It may be much poorer than average and sometimes drops below the acceptable level – case study is provided. Even if the links are fully operational, due to the rerouting the QoS may deteriorate in the case of cable fault in another part of the World. The research is based mainly on data taken from IEPM (Internet End-to-end Performance Measurement) database.

Keywords: QoS measurement, large scale networks, performance, reliability.

1 Introduction

It is well known that the term “quality of service” is used in many meanings ranging from the user’s qualitative perception of the service to a set of quantitative connection parameters (RTT, jitter, throughput, packet loss rate) necessary to achieve particular service quality. In the paper we will mostly use the second meaning of the term. This meaning is consistent with IETF approach presented in RFC 2386 [4].

It is relatively easy to provide high quality of service in short-distance, local connections. Small number of network devices, usually homogenous and managed by single service provider facilitates optimization with such techniques as MPLS, RSVP, header compression, TCP and web acceleration, redundancy and so on.

On the other hand accomplishing high QoS in intercontinental connections is particularly tough problem. First of all packet delay is much longer, with approximately 4.5 μ s per each kilometer of fibre and more network devices introducing delays. Long-distance communication channels are usually shared by many users and institutions. They are heterogeneous: they carry different types of traffic: data, phone calls, ATM (Automated Teller Machine) transactions, they consist of diverse network devices. All that means:

- transient nature of IP behavior,
- more complex management,

- more bottlenecks
- more points of failure,
- more difficult and time consuming diagnosis and fixing after a failure.

Repairing intercontinental submarine cable after breakage (especially in the case of earthquake) may take weeks and even months. Achieving high reliability with redundancy is very expensive. In addition network resources are administered by various network providers, with diverse and occasionally conflicting objectives.

There are many services for Internet performance measurement¹. Some of them are integrated with databases with present and past measurements. The example is IEPM PingER, a service monitoring performance of Internet links, developed at SLAC (Stanford Linear Accelerator Center) and operating since 1995 (with data stored since 1998). Monitoring is based on more than 300 distributed hosts. Hosts send periodically pings for each tested connection. The measurement results are written to database. PingER database is used in the next parts of the paper.

2 Long-Term QoS Changes in Transatlantic Connections

2.1 Transatlantic Cables Deployment

Long distance connections are based on telecommunication cables and satellite transceivers. Satellite transmission QoS is relatively low, with significant delay and small bandwidth (e.g. one way signal propagation via geostationary satellite at 36 000 km altitude takes 260 ms [6]) and shorter design life (10-15 years compared to about 25 years for cable). So long distance communication is based mainly on cables. It may be assumed that 95% of all intercontinental links are based on cables.

Transoceanic cables are used since 1858. In 1988 first fibre-optic cable TAT-8 had been laid in North Atlantic with capacity of 280 Mbit/s. Since that year many

Table 1. Transatlantic (North America-Europe) cables operated in 1998 [www.iscpc.org]

Year of operation start	Name	Bandwidth [Gbit/s]
1988	TAT-8	0.28
1989	PTAT-1	0.42
1992	TAT-9	0.56
1992	TAT-10	0.56
1993	TAT-11	0.56
1994	CANTAT 3	2.5
1994	Columbus-II	1.68
1996	TAT-12/13	15
1998	Gemini	115
1998	AC1	40

¹ For example: <http://www.internettrafficreport.com/> or <http://visualroute.visualware.com/>

Table 2. New transatlantic (North America-Europe) cables deployment in 1999-2008 [www.iscpc.org]

Year of operation start	Name	Bandwidth [Tbit/s]
1999	Columbus III	0.04
2000	AC-2	0.64
2001	Hibernia Atlantic	0.16 (lit) 1.92 (designed)
2001	TAT-14	1.87 (lit) 3.2 (designed)
2001	VSNL Transatlantic	5.12
2001	FLAG Atlantic 1	4.8
2003	Apollo	3.2 (designed)

transoceanic cables were deployed in North Atlantic with growing capacity per cable.

It may be assumed that total capacity of North Atlantic cables operated in 1998 was equal roughly to 150 Gbit/s (table 1). Approximate, total bandwidth of several North Atlantic cables deployed from 1999 to 2008 is 15 Tbit/s (table 2). Data are based on ICPC (International Cable Protection Committee) resources.

So we have approximately 100-fold increase in total submarine connections bandwidth in 10 years time. This is consistent with the rate in which DWDM fibre links are improved.

2.2 Lasting Trends in Performance

Introduction. IEPM database² preserves Internet performance data for many connections (pairs of monitoring sites) throughout the World [3]. The number of connections monitored at the given date varies in time. In the period 1998-2008 among 47 and 154 connections between Europe and USA were monitored. The connections are tested with packet sizes of 100 and 1000 bytes. In the next part average values of QoS parameters of the connections with 100 bytes per packet are provided.

Average, yearly (1998-2008) values of QoS parameters for Europe to USA connections are presented in table 3. The communication channels are asymmetric, so the parameters for reverse direction (USA to Europe connections) are different (table 4). Generally USA to Europe parameters are worse than Europe to USA. For example, more of the time, Europe to USA throughput is greater than USA to Europe (figure 1). The difference is in the range of about 140-600 kbit/s or 20-25%.

All analyzed QoS parameters have improved in a given period of time. Nevertheless the improvement pace is not smooth and rather slow compared to other computer and network performance indicators (e.g. bandwidth of Ethernet or Moore's Law).

² <http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl>

Table 3. Long-term changes of QoS on transatlantic (Europe to USA) connections

Year	RTT [ms]	Jitter [ms]	Throughput [kbit/s]	Packet loss [%]
1998	219	0	530	5.0
1999	213*	160	531	4.3
2000	171	143	840	1.8
2001	153	3.7	1385	0.8
2002	151	149	1525	1.0
2003	150	227	1788	0.4
2004	146	82	1899	0.6
2005	155	348	2472	1.6
2006	155	84	2271	1.9
2007	154	1.9	1958	5.5
2008	154	2.1	2985	0.7

* Value calculated after excluding 2 anomalous numbers of RTT from the IEPM PingER database. Without excluding the numbers RTT for 1999 would be at the level of 9120 ms.

Table 4. Long-term changes of QoS on transatlantic (USA to Europe) connections

Year	RTT [ms]	Jitter [ms]	Throughput [kbit/s]	Packet loss [%]
1998	237	0.0	391	3.7
1999	233	853	429	5.1
2000	206	154	609	2.8
2001	252	69	1026	1.2
2002	227	46	1294	0.9
2003	183	3.8	1712	0.8
2004	169	3.5	2622	0.6
2005	158	66	2523	0.4
2006	171	1.6	2301	0.1
2007	172	1.4	1716	0.2
2008	177	1.2	2371	0.5

“(...) it would be easy to telegraph from Ireland to Newfoundland at a speed of at least from eight to ten words per minute.” Samuel F.B. Morse, 1856

Throughput. Throughput is a fraction of bandwidth. In the given period of time we’ve observed roughly 100-fold increase in total available bandwidth between North America and Europe (see 2.1). On the other hand, average throughput enhance is much smaller and irregular. In 1998 and 1999 throughput remained at the level of about 400-500 kbit/s. The following years brought relatively low increase to 2.4-3.0 Mbit/s (figure 1). So we’ve observed about 6-fold increase. This 2.4-3.0 Mbit/s throughput is enough for mail, web access and several VoIP channels but is much below threshold necessary for video applications with PAL-MPEG2 coding.

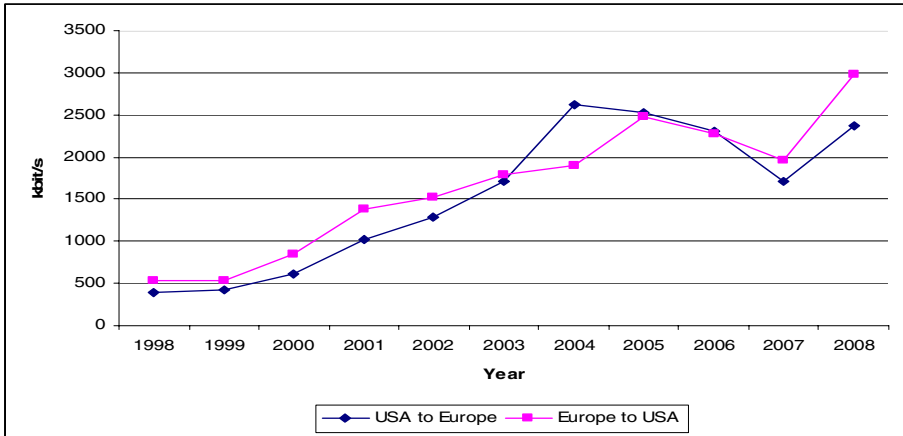


Fig. 1. Long-term changes of average throughput on transatlantic connections

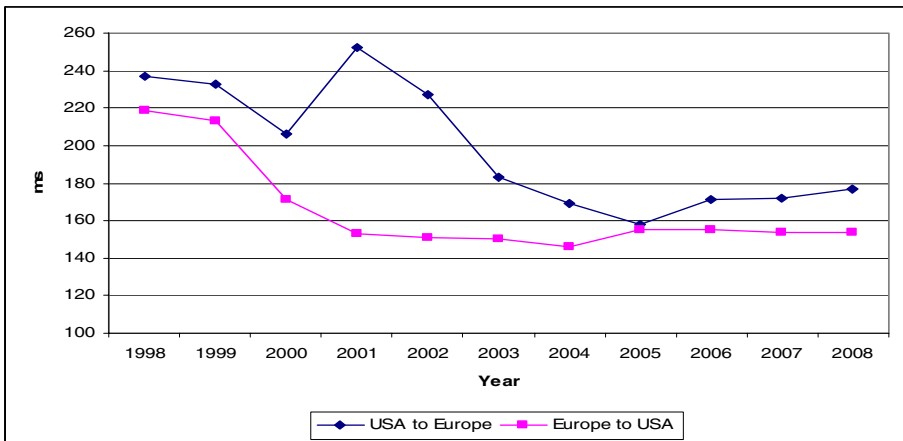


Fig. 2. Long-term changes of average RTT on transatlantic connections

RTT and Jitter. RTT is total propagation time needed to transfer IP datagram from one host to another and back. It includes delays in a fibre and all intermediate network devices (routers, switches, repeaters, transmit and receive terminals). Typical length of North Atlantic cable is between 6000 and 7500 km. The speed of infrared waves in a fibre³ is close to 220 000 km/s, so the signal travels one way through the cable in 27-34 ms. In the worst case transmit terminal in submarine optical fibre system adds 13 ms delay and receive terminal 10 ms delay [6].

In 1998 average RTT remained at the level of roughly 220-240 ms. It decreased to the level of 150-180 ms in the first 4 years of 2000's and stayed at this level to 2008 with minor fluctuations (figure 2). So, one-way datagram delay is 75-90 ms. This

³ In fact the speed is dependent on wavelength but the variations are negligible in this case.

value is well below ITU-T 200 ms threshold for absolute (mouth to ear) delay in VoIP applications in which users are very satisfied [6]. On the other hand the requirements for control and feedback in medical operations are hardly satisfied (it is assumed that one-way delay should be less than 80 ms in this case [5]).

Assuming that average one way delay between North America and Europe (in 2008) is in the range 75-90 ms, the cable propagation time (27-34 ms) is about 1/3 of the total delay and 2/3 is introduced by network devices.

The greatest fluctuations of average values are observable in jitter. It changes from below 2 ms up to several hundreds ms.

Packet Loss Rate. Average packet loss rate slightly improves in time. In 1998 it was at the level of 4-5%. It decreased to the level below 1% and stays at this level with some temporary deteriorations (e.g. elevated 5.5% rate in 2007 for Europe to USA connections). Packet loss rate below 1% is recognized as acceptable for interactive applications such as VoIP⁴.

3 Short-Term Fluctuations in Normal Operation

Section 4 presents large changes to average daily QoS in the case of submarine cable fault. Before we present and analyze the data we should see how QoS parameters fluctuate (table 5) during normal operation time (without cable faults). The same

Table 5. Short-term fluctuations of QoS on exemplary Europe to India (n2.cern.ch to n1.cnieds.bangalore.in) connection during normal operation (16-28 January 2008)

Date	RTT [ms]	Jitter [ms]	Throughput [kbit/s]	Packet loss [%]
Dec 2007 average	184	7.9	1244	0.3
Jan 16	185	7.7	1027	0.4
Jan 17	180	6.8	564	1.3
Jan 18	193	9.2	1910	0.0
Jan 19	194	10.3	1901	0.0
Jan 20	179	7.1	2059	0.0
Jan 21	194	9.4	1022	0.3
Jan 22	205	9.2	1801	0.0
Jan 23	213	16.0	893	0.4
Jan 24	201	10.3	1333	0.2
Jan 25	197	10.3	461	1.7
Jan 26*	-	-	-	-
Jan 27	187	6.3	811	0.6
Jan 28	193	8.0	1394	0.2
Jan 29	190	10.4	524	1.4

* Data for January 26 are not available in the PingER database.

⁴ Assuming other measures (delay, jitter) are at the satisfactory levels. Additionally, voice quality is related to the character (random or bursty) of the losses.

exemplary Europe to India⁵ connection is evaluated in both cases (between n2.cern.ch and n1.cnieds.bangalore.in).

Fluctuations of QoS in the given period of regular operation time are visible. Minor fluctuations are observable in RTT and packet loss rate. RTT changes in the range of 179-213 ms. Packet loss rate do not exceeds 1.7%. Jitter stays below 16 ms (this is 2x more than average December 2007 jitter). Highest increases and decreases are noticeable in throughput. On January 20 it reaches more than 2 Mbit/s (166% of December 2007 average), but on January 25 it drops below 0.5 Mbit/s (37% of December 2007 average).

On the sidelines we may analyze weekdays fluctuations. Some minor fluctuations are visible in the average QoS of particular days of week. Throughout a given period of time (e.g. year) Saturdays and Sundays have highest average QoS and Mondays have lowest (it may be seen also in table 5: in the period 16-28 January 2008, highest throughput and lowest RTT are on Sunday, January 20). Nevertheless in a given week of the year it is not possible to predict a day with the highest QoS level.

4 Short-Term Fluctuations in the Case of Cable Fault

4.1 Submarine Cable Faults

Network reliability and performance is disrupted by intentional and unintentional factors: malicious software, spam, hackers and disasters (e.g. earthquakes). First 3 factors are widespread. Damages imposed by the factors are usually restricted to single services and are short-lived. On the other hand disasters are uncommon but their impact is long-lived. They disrupt Internet services, telephone calls and ATM transactions.

Intercontinental cable faults⁶ are relatively infrequent (in 2003 annual fault rate was at the level of one fault per 10000 km of cable [7]). In the case of faults channel redundancy plays an important role. In the Atlantic, cable breaks happen repeatedly (more than 50 cable repairs are yearly in the Atlantic) but due to the high level of redundancy (about 20 cables connect nowadays Europe with North America) they are almost invisible to end users. On the other hand disasters on Mediterranean inflict big impact on Internet services. Submarine cables are prone to being affected by earthquakes, storms, fishing and anchors. 70% of faults are attributed to human activity with fishing as the major cause [7]. Usually, earthquake extent of the damage is much greater than anchor cut, the cable may suffer several breaks. The severed ends could be buried by deep-sea landslides or washed kilometers from their previous positions. It may take many days to just find the cable and months to full service restoration.

In November 2003 TAT-14 (USA-Europe) cable fault occurred. Many Internet service providers in UK experienced some problems. As TAT-14 is a dual, bi-directional

⁵ India is an IT outsourcing centre, with Bangalore as India's Silicon Valley, so the connections to India's networks are particularly important for US and European businesses.

⁶ Satellites are also vulnerable, they may move away from orbit or may collide (e.g. on February 11 2009 Iridium Satellite collided with Russian Cosmos 2251 satellite, an incident resulted in limited disruptions of Iridium service).

ring of cable, a single serious fault should not be enough to break it, as traffic would still be able to flow between the countries on the ring. Unfortunately, a part of the cable near the USA coast had already suffered a technical fault few days earlier, which meant there was no built-in redundancy.

In May 2003 6.8 magnitude earthquake with epicenter near Algiers damaged five submarine cables. All Algerian voice, mobile and Internet traffic was disrupted. The last repair completed only 6 weeks after the earthquake.

In June 2005 SMW 3 cable had been cut off Karachi. Pakistan lost all terrestrial Internet connectivity. The outage of services lasted 12 days.

On December 17, 2006, CANTAT-3 cable connecting Iceland with Europe and Canada was damaged. Most notable effects of the event was a temporary shut-down of data-communications by Iceland's universities and hospitals which rely exclusively on CANTAT-3's services. It took more than 7 months, until July 29, 2007 before service was fully restored.

On December 26 2006 a 7.1-magnitude Hengchun earthquake south of Taiwan knocked seven submarine cables out of service, impairing communications from North America to China, Taiwan, Japan and Korea as well as inside North and Southeast Asia. Cables accounting for 90% of telecommunications capacity of the region had been broken. It took 49 days to repair all the cables.

On January 30 2008 a series of accidents with Mediterranean cables started (p. 4.2).

On December 19 2008 five submarine cables (including SMW 3, SMW 4, FLAG, Seabone) had been cut near Sicily due to 5.3 magnitude quake in the central Mediterranean. The cut disrupted Internet and telephone services in the region and in parts of the Middle East and South Asia [2].

4.2 Case Study

Introduction. January 2008 Mediterranean accident is an interesting research subject from many points of view. It consisted of several events. It occurred in the area of high traffic and relatively small connection redundancy. It showed that cable faults have high and distributed impact on Internet performance. It demonstrated several problems with current Internet infrastructure. The accident is well documented. News networks provided many reports. IEPM preserved data (RTT, throughput, jitter) on net performance during the accident. RIPE [9] (Réseaux IP Européens) and Renesys [8] researched IP route changes.

Accident Timetable. The accident was a series of events. Cables from Europe to Middle East and Asia were affected: January 30, 2008 about 4:30⁷ SMW 4 cable (deployed in 2005, total capacity 1.28 Tbit/s) near Alexandria was damaged, January 30 about 8:00 FLAG cable (deployed in 1997, total capacity 10 Gbit/s) near Alexandria was damaged. The two cables carry about 70% of the traffic between Europe and the Middle East. February 1 about 6:00 FALCON cable near Dubai was damaged, February 8 SMW 4 cable was repaired, February 9 FLAG cable was repaired, February 10 FALCON cable was repaired [9]. It is assumed that the cables were damaged by anchors.

⁷ All times are UTC.

Table 6. Fluctuations of QoS on exemplary Europe to India (n2.cern.ch to n1.cnieds.bangalore.in) connection during cable fault

Date	RTT [ms]	Jitter [ms]	Throughput [kbit/s]	Packet loss [%]
Dec 2007 average	184	7.9	1244	0.3
Jan 28	193	8.0	1394	0.2
Jan 29	190	10.4	524	1.4
Jan 30*	-	-	-	-
Jan 31	614	6.8	40	23.0
Feb 1	563	66.5	38	30.9
Feb 2	502	13.0	54	18.4
Feb 3	479	6.8	131	3.5
Feb 4	526	14.7	106	4.4
Feb 5	658	8.5	561	0.0
Feb 6	619	22.1	434	0.2
Feb 7	486	6.7	194	1.5
Feb 8	369	5.9	564	0.3
Feb 9	184	1.4	786	0.6
Feb 10	193	8.7	874	0.5
Feb 11	195	10.6	1897	0.0

* Data for January 30 are not available in the PingER database.

IP Rerouting. Three general types of effects to IP routing are possible in the case of submarine cable cut: immediate loss of connection between some networks after the failure, much smaller number of available AS (Autonomous System) paths and likely rerouting with a use of backup paths. These backup paths are longer and offer poorer performance. A good indication of the impact is the number of IP address network prefixes, that are announced in BGP messages. If prefix to a given network is not announced to routing peers then the network is not reachable. In some countries (Egypt, Kuwait, Sudan), immediately after the failure, more than 30% of the prefixes were removed from BGP announcements. The total number of AS paths decreased and many networks disappeared from the routing tables. The number of changes in AS paths rapidly increased in the region. Not more than 10% of paths change daily in normal operation routing mode. This measure grew, in the region, to above 60% on January 30. This higher than normal and fluctuating (between 10–30%) paths change percentage persisted up to February 18. Additionally, average AS path length (the number of different ASes between two distant networks) increased slightly, around the time period of the cuts from 5.5 to about 6.

This was caused by rerouting of some connections. Europe-Asia connections were rerouted through the SMW 3 (deployed in 1999, total capacity 20 Gbit/s) cable or fibres taking the way around the globe (Europe-USA-Asia). Due to the limited bandwidth and traffic increase on these new routes it was difficult to quickly converge routing tables on alternate topologies [9]. This is an effect of distance-vector algorithm for BGP routing optimization. It will be shown (table 7) that traffic rerouting had significant impact not only on rerouted connections but also on some of

the USA to Asia connections. IP routes modifications resulted directly in significant changes of RTT, throughput and packet loss rate. Parameters returned fully to their values from before accident only several days after the time in which all repairs were completed.

Fluctuations of performance parameters on exemplary Europe to India connection (between n2.cern.ch and n1.cnieds.bangalore.in) are presented in the table 6 and discussed in the next subsections. It is evident that the connection was directly hit by the cable cut.

Since the incident had indirect impact on QoS in other parts of the World, network parameters on exemplary USA to India connection (between n8.doe.gov and n1.cdacmumbai.in) are presented in the table 7 and discussed in the next subsections.

Table 7. Fluctuations of QoS on exemplary USA to India (n8.doe.gov to n1.cdacmumbai.in) connection during cable fault

Date	RTT [ms]	Jitter [ms]	Throughput [kbit/s]	Packet loss [%]
Dec 2007 average	257	7.5	223	4.2
Jan 28	255	4.1	203	5.1
Jan 29	254	4.4	224	4.2
Jan 30*	456	12.5	39	43.3
Jan 31	515	18.6	34	45.2
Feb 1	401	18.7	53	30.5
Feb 2	421	14.9	58	22.7
Feb 3	382	15	128	5.7
Feb 4	369	13.8	98	10.4
Feb 5	357	9.2	145	5.1
Feb 6	348	6.2	114	8.7
Feb 7	356	11.5	79	17.2
Feb 8	362	24.1	110	8.6
Feb 9	387	32.9	168	3.2
Feb 10	320	15.8	203	3.2
Feb 11	257	2.7	220	4.3

Throughput. Average monthly throughput (assuming TCP connections with 100 bytes per packet) for Europe to India connection was in December 2007 at the level of 1244 kbit/s. It rapidly decreased on January 31 to 40 kbit/s. That means 30-fold worsening. It should be observed that this is unacceptable throughput level for VoIP services. Throughput remained below 1000 kbit/s up to February 10 (figure 3).

Similarly, average monthly throughput for USA to India connection was in December 2007 at the level of 223 kbit/s. It rapidly decreased on January 30 to 39 kbit/s. Throughput remained well below 200 kbit/s up to February 9 with temporary improvement (to 145 kbit/s) on February 5 (figure 3).

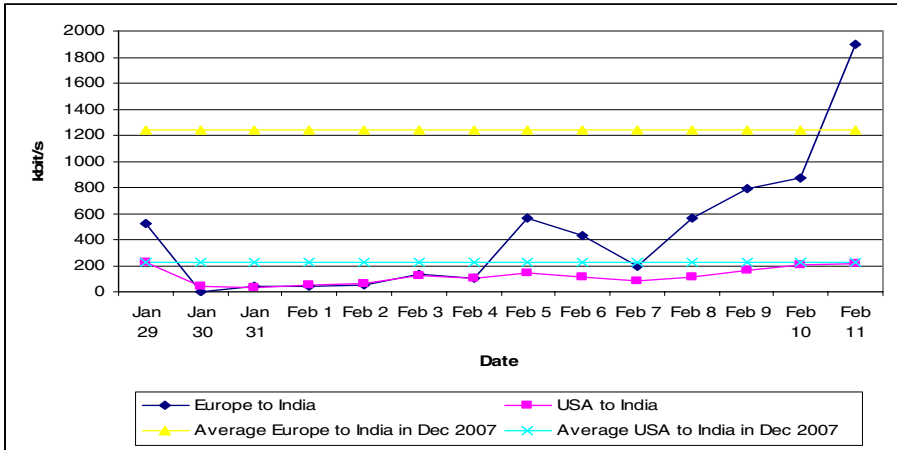


Fig. 3. Fluctuations of throughput during cable fault

RTT and Jitter. Average monthly RTT in December 2007, for exemplary Europe to India connection, was 184 ms. Connections between Europe and India had been significantly affected: RTT increased 3x on January 31 to above 600 ms. Changes were inflicted by traffic rerouting through USA networks which started to carry additional traffic. In consequence connections between USA and India had been affected too: average RTT (USA to India) increased 2x on January 31 from 257 ms (average in December 2007) to 515 ms. RTT increased due to longer routes, more routers and longer queues in routers, which have to carry additional traffic. Average RTTs started to improve after January 31 and returned with fluctuations (especially on February 4-5 on Europe to India connection) to their values from before the accident after 12 days (figure 4).

Average monthly jitter for Europe to India connection was (in December 2007) at the level of 7.9 ms. It slightly increased during first 2 days to 10.4 ms and sharply increased on February 1 to above 66 ms. Next days it started to improve with oscillations. Average monthly jitter for USA to India connections was in December 2007 at the level of 7.5 ms. It increased during first 2 days to 19 ms. The highest jitter 24–33 ms appeared on February 8 and 9, at the time of cables repairment.

Packet Loss Rate. Average monthly packet loss for Europe to India connection was in December 2007 at the level of 0.3%. It increased after cable fault up to 30.9%. This level of packet loss means total unavailability of interactive services including e.g. control and feedback for remote medical operations [5]. Average monthly packet loss for USA to India connections was in December 2007 at the level of 4.2%. It increased on January 31 to 45%. Packet loss rate temporarily recovered in next few days. On February 3 we observe decrease for Europe to India and USA to India connections. In both cases the level not greater than about 6% has been achieved. In the next few days it fluctuated between 0 and about 20% and returned to its values from before the accident on February 9.

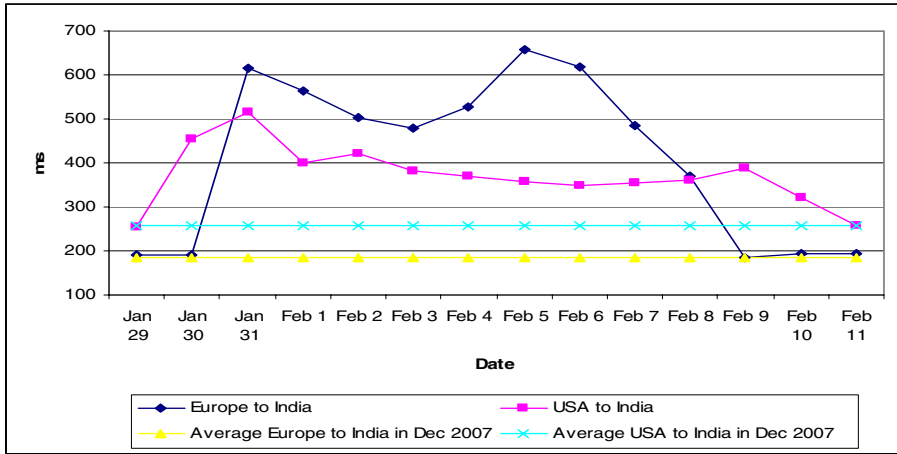


Fig. 4. Fluctuations of RTT during cable fault

Cable Cut Impact on QoS. Internet survived the accident. Nevertheless the disaster’s impact on QoS was: significant, long-lived and widespread (cable cut near Alexandria degenerated USA to Asia connections which do not use the cable). The performance parameters during the accident decreased to unacceptable (for interactive applications) levels. Internet performance have been changing in unpredictable way (it is seen for example on throughput and RTT data). The effects of submarine cable cut accident are notably different from the effects of hacker attack on Internet server. The analyzed accident was not an exception. Similar accidents in the future should be expected. Generally we are not able to predict time and place of the accidents (e.g. earthquakes are hardly predictable) [1].

5 Conclusions

Intercontinental connections performance is upgrading very slowly and irregularly. Compound growth of throughput for exemplary transatlantic connections in 10 years is just 6-fold on average. The increase rate is lagging behind other computer and network performance indicators. At the same 10-years time total North Atlantic bandwidth increased approximately 100 times. This is related to fiber transmission capacity and DWDM link speed, which grow by a factor of about 200 in the decade. Similarly, according to Moore’s Law computer power increase in the same period is 100-fold. The router capacity, which takes advantage of the Moore’s Law increases at approximately the same rate. Hard disk data areal density grows also at the similar rate. Ethernet bandwidth is growing exponentially from 10 Mbit/s in 1989 to 10 Gbit/s in 2002 and 100 Gbit/s in 2010 (2010 is feasible year of the new standard ratification) – this means 10000-fold increase in 20 years. Discrepancy between different factors influencing IT performance is visible.

QoS of intercontinental connections is hardly predictable. Day to day fluctuations are large, even in normal operation time. Irregular disasters with submarine cable

faults make this predictability more complicated. In the case of such fault temporary level of performance may rapidly drop to unacceptable value (e.g. throughput 40 times lower than average). Some cable faults have high and distributed impact on Internet performance. Significant QoS deterioration may last days and sometimes weeks and months. Such disasters have impact on many telecommunication services: email transfers, web access, telephone calls and ATM transactions. There are differences between cable fault and intentional attack consequences. In a typical case damages imposed by hackers and malicious software are usually short-lived and restricted to single computer or service.

It is clear that many things should be done to improve QoS on long-distance connections. The changes ought to be implemented in networks as well as in hosts and applications.

Network (fibre) redundancy should be carefully planned at every infrastructure level. For example, for end user it is pointless to use two ISPs if both utilize the same international cable. Of course more cables are needed. Their location should be better planned. Existing cables should be upgraded so that they are operating at a percent of their potential capacities, leaving plenty of room not only for future traffic growth but also for rerouted (in the case of disaster) traffic. BGP protocol should be upgraded or replaced with completely new one EGP protocol. Routing table convergence time should become important optimization criterion. More efficient implementations of TCP (MulTCP, HighSpeed TCP) should be integrated with common operating systems.

Globally used services should be based on data mirroring, caching proxies, CDNs (Content Delivery Networks). Applications, which vary in their QoS requirements should be better profiled before determining appropriate classification and routing treatment. Both versions of IP are ready for such data classification. Routers should be aware and able to carry differentiated traffic. To save bandwidth real time applications should be based on more efficient codecs. RTP/UDP/IP header compression should be broadly utilized.

Of course it is easy to recommend actions, procedures, modifications and much harder to apply them. Suggested improvements are not easy to implement but many of the techniques, algorithms, protocols and their implementations are already available.

References

1. Bilski, T.: Disaster's Impact on Internet Performance - Case Study. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2009. CCIS, vol. 39, pp. 210–217. Springer, Heidelberg (2009)
2. Cottrell, L.: Effects of Mediterranean Fibre Cuts December 2008, SLAC (2009), <https://confluence.slac.stanford.edu/display/IEPM/Effects+of+Mediterranean+Fibre+Cuts+December+2008>
3. Cottrell, L., Matthews, W., Logg, C.: Tutorial on Internet Monitoring and PingER at SLAC. In: SLAC 2007 (2007), <http://www.slac.stanford.edu/comp/net/wanmon/tutorial.html>
4. Crawley, E., Nair, R., Rajagopalan, B., Sandick, H.: RFC 2386 – A Framework for QoS-based Routing in the Internet. Network Working Group (1998)

5. Gutierrez, D., Shah, A., Harris, D.A.: Performance of Remote Anatomy and Surgical Training Applications Under Varied Network Conditions. In: Barker, P., Rebelsky, S. (eds.) Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002, pp. 662–667. AACE, Chesapeake (2002),
<http://havnet.stanford.edu/pdfs/ed-media.pdf>
6. ITU-T Recommendation G.114. One-way transmission time, ITU-T 05/2003
7. Kordahi, M.E., Shapiro, S.: Worldwide Trends in Submarine Cable System Faults. In: SubOptic 2004 (2004),
<http://www.suboptic.org/Viewdocument.aspx?id=381>
8. Popescu, A., Premore, B., Zmijewski, E.: Impact of the Middle East Cable Breaks. A Global BGP Perspective, Renesys Corp. (2008),
<http://www.renesys.com/tech/presentations/pdf/nanog42-lightning.pdf>
9. Wilhelm, R., Buckridge, C. (eds.): Mediterranean Fibre Cable Cut – a RIPE NCC Analysis, RIPE (2008),
<http://www.ripe.net/projects/reports/2008cable-cut/index.html>

Performance-Adaptive Prediction-Based Transport Control over Dedicated Links

Xukang Lu¹, Qishi Wu¹, Nageswara S.V. Rao², and Zongmin Wang³

¹ Dept of Computer Science, University of Memphis, Memphis, TN 38142, USA
{xlv,qishiwu}@memphis.edu

² Computer Sci. Math. Div., Oak Ridge National Lab., Oak Ridge, TN 37831, USA
raons@ornl.gov

³ Henan Key Lab On Info. Net., Zhengzhou Univ., Zhengzhou, Henan 450052, China
zmwang@zzu.edu.cn

Abstract. Several research and production networks now provide multiple Gbps dedicated connections to meet the demands of large data transfers over wide-area networks. End users, however, have not been able to see corresponding increase in application goodputs mainly because (i) such rates have pushed the bottleneck from the network to the end system, and (ii) the traditional transport methods are not optimized for handling host dynamics. Due to the sharing with unknown background workloads, the data receiver oftentimes lacks sufficient system resources to process packets arriving from high-speed dedicated links, therefore leading to significant packet drops at the end system. We propose a rigorous design approach for a new class of transport protocols that explicitly account for the dynamics of the running environment to maximize application goodputs over dedicated connections. The control strategy of the proposed transport method combines two aspects: (i) the receiving bottleneck rate is predicted based on performance modeling, and (ii) the sending rate is stabilized at the estimated bottleneck rate based on stochastic approximation. We test the proposed method on a local dedicated connection and the experimental results illustrate its superior performance over existing methods.

Keywords: Transport control, dedicated networks, performance modeling.

1 Introduction

Many large-scale scientific, engineering, and e-commerce applications require the rapid transfer of vast amounts of data on the order of terabytes or petabytes. Efforts to improve the data transfer performance in the shared Internet met little success due to the variable limited available bandwidth in response to cross traffic. Dedicated networks provisioning multiple Gbps connections have been recognized to be a promising solution and a number of high-performance network initiatives are currently underway including Dynamic Resource Allocation via

GMPLS Optical Networks (DRAGON) [1], UltraScience Net (USN) [17], Circuit-switched High speed End-to-End Transport Architecture (CHEETAH) [20], and others.

However, end users have not been able to see corresponding goodput¹ increase in their applications mainly because (i) such rates have pushed the bottleneck from the network to the end system, and (ii) the traditional transport methods are not optimized for handling host dynamics. Due to the lack of a system-wide advance reservation scheme, the data receiver running in a shared computing environment with other resource-demanding workloads oftentimes could not obtain sufficient system resources to process packets arriving from high-speed dedicated links, therefore leading to significant packet drops at the end system.

The current research efforts on transport protocol design are mainly focused on TCP enhancements and rate-based application-level protocols over UDP. The widely deployed TCP, which has been proved to be remarkably successful in the Internet, is not adequate to achieve high goodput in wide-area dedicated networks because the Additive Increase Multiplicative Decrease (AIMD)-based congestion control algorithm is not well suited for links with high Bandwidth Delay Product (BDP). In TCP, packet loss is detected either by timeout of an unacknowledged segment or several duplicated acknowledgements. If packet loss is caused by network congestion, TCP is able to achieve a reasonable link utilization. However, many observations have shown that packet loss is a poor indicator of network congestion, especially in high-speed dedicated networks where congestion has been pushed to the end system. Various TCP enhancements have been proposed to improve throughput performance, including TCP vegas [5,16], Scalable TCP [12], High Speed TCP for large congestion windows [14], XCP (eXplicit Control Protocol) [11], and many others [8]. Diverging from TCP's AIMD control, a number of UDP-based high-performance transport protocols use non-AIMD rate control to overcome TCP's throughput limitation for high BDP networks. These protocols include Hurricane [18], SAUBUL (Simple Available Bandwidth Utilization Library)/UDT (UDP-based Data Transfer) [9], FRTP (Fixed Rate Transport Protocol) [19], RBUDP (Reliable Blast UDP)/LambdaStream [10], and Tsunami [2].

However, the main design goal of the aforementioned transport methods based on either TCP or UDP is still to address the congestion over network links, not to account for the dynamics of the end system. As a matter of fact, besides processing power, many other host factors including NIC-system-application interactions, memory/buffer management, CPU scheduling, and disk I/O, collectively contribute to the complex end-system dynamics and play a significant role in determining the application goodput in high-speed dedicated networks. Therefore, to maximize application goodputs, transport protocols need to incorporate a performance-adaptive mechanism through which the data sender and receiver could suitably adjust their sending and receiving activities in response to the system dynamics.

¹ Goodput only counts the user payload and is equivalent in value to throughput if packet duplicates and protocol headers are negligible.

The goal of our work is to present Performance-Adaptive Prediction-based Transport Control (PAPTC) that explicitly accounts for the dynamics of the end system to maximize application goodputs over dedicated connections. With rigorous design and careful analysis, the control strategy of PAPTC combines two aspects: (i) the receiving bottleneck rate is predicted based on performance modeling, and (ii) the sending rate is stabilized at the estimated bottleneck rate based on a stochastic approximation (SA) method. We construct a mathematical model for the data receiving process and employ an autoregressive method to predict the receiving bottleneck rate, which is sent back to the sender for rate control. To account for both network and host dynamics and achieve quick convergence, we adjust the source rate for goodput stabilization at the estimated receiving bottleneck rate using the Robbins-Monro SA algorithm: the source rate is continuously adjusted to match the bottleneck receiving rate at a strategically selected interval. We test the proposed method on a local dedicated connection and the experimental results illustrate its superior performance over existing methods.

The rest of the paper is organized as follows. In Section 2, we briefly outline the framework of PAPTC structure. In Section 3, we present a performance model for the data receiver, and in Section 4, we describe the rate control algorithm for the data sender. The implementation details and experimental results are provided in Section 5.

2 Framework of PAPTC Structure

PAPTC employs a UDP-based transport control structure for disk-to-disk data transfer as shown in Fig. 1. The sender (source) reads data sequentially from its local storage device as a set of UDP datagrams of *Maximum Datagram Size* (MDS), each of which is assigned a unique continuous sequence number and loaded into the sender buffer. The source sending rate $r_S(t)$ at time t is regulated by a pair of congestion window $W(t)$ and sleep or idle time (i.e. inter-window delay) $T(t)$. The receiver (destination) accepts incoming datagrams in the order of their arrival and keeps track of the datagram sequence numbers in a checklist. The received datagrams are immediately forwarded to a disk I/O module that handles datagram reordering if necessary and writes them to the disk in order in the background. Based on the status of the datagram checklist, an either positive or negative acknowledgment (ACK) of lost datagrams during an interval $I(t)$ is generated and sent periodically to the sender for retransmission.

As shown in Fig. 1, the data flow moves from source to destination along the solid lines and the acknowledgment feedback follows the dotted lines from destination to source. In this transport structure, there are two control operations represented by two shaded elliptic boxes: (a) source rate control through idle time and (b) ACK event interval control. The transport performance over high-speed dedicated channels critically depends on the strategies implemented for these two control operations. Many transport control protocols send a positive acknowledgment for a received data packet, which is necessary for shared lossy

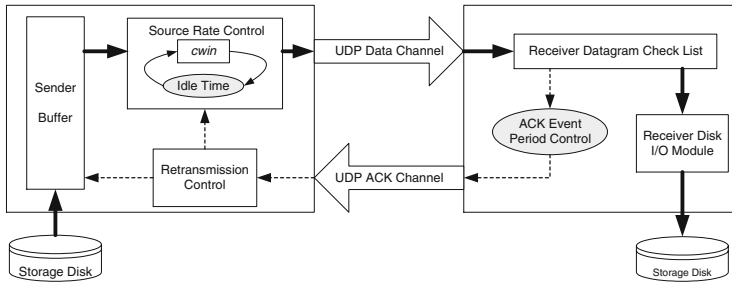


Fig. 1. Transport control structure for disk-to-disk data transfer

links in Internet environments. However, dedicated channels usually provide very reliable connections where packet loss rarely occurs. At high data rates, generating and sending acknowledgments at the receiver consumes CPU time and may interfere with the host receiving process. Similarly, accepting and processing acknowledgments at the sender may also affect the host sending process. To achieve peak performance over dedicated channels, we employ a mixed acknowledgment mechanism that sends an either positive or negative acknowledgment after a carefully selected period of time. An appropriate delay time of mixed acknowledgments is adaptively determined for network connections based on link and host properties.

3 Performance Model for Data Receiver

We present an analytical study on the impact of system properties on the performance of transport protocols. To instantiate the analysis, we consider the Linux kernel.

3.1 Packet Processing Issues

For convenience, we plot in Fig. 2 an overview of Linux packet processing that involves the NIC hardware, device drive, kernel protocol stack, and application [7]. When a new packet arrives, the NIC generates an interrupt and the packet is put into the kernel buffer by the card DMA engine. In general, heavily engaging the CPU in other compute-bound tasks during an interrupt may severely hinder a running process. To avoid flooding the host system with too many interrupts, the interrupt coalescence scheme collects multiple packets and generates one single interrupt for them, therefore reducing the amount of time that the CPU would otherwise have to spend on context switching to serve multiple interrupts. The Linux kernel uses *sk_buff* structure to hold any single packet. The pointers of *sk_buff* are held in a ring buffer in the kernel memory and manipulated through the network stack. If there are no free pointers in the ring buffer, incoming packets will be dropped by the kernel silently. From the ring buffer, the packets are delivered to the corresponding receiving function of the IP layer, which examines

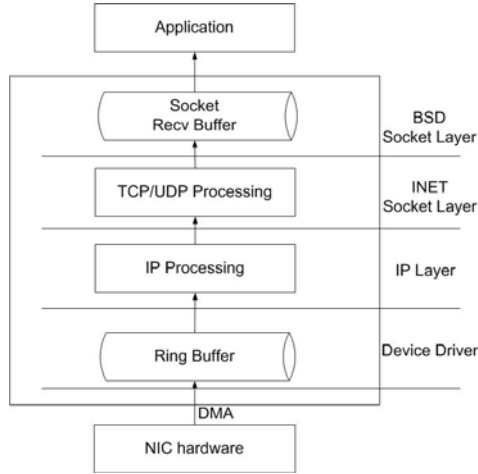


Fig. 2. Packet processing flow in Linux

the packets for errors and then forwards them up to the INET Socket layer (such as TCP or UDP), which in turn checks for errors and copies the packets into the socket receive buffer. Then, the waiting application wakes up and returns from a corresponding receive system call that copies the data from the kernel into the application buffer. The flow control mechanism of TCP is implemented to avoid packet drops in the receive buffer. However, the UDP receive buffer might be overflowed if the packet receiving process can not acquire enough CPU cycles to consume the data in the buffer due to CPU contention. In this case, all incoming packets are discarded, hence wasting the protocol processing resources and impairing the application performance.

The Linux packet processing flow shows that packet drops by the kernel could happen in either the ring buffer, or the socket receive buffer, or both. Since the data receiving process has a lower priority than the packet processing by the kernel and the Interrupt Service Routine (ISR), packets are more likely to drop in the socket receive buffer. Although UDP is buffered on both the sender and receiver sides, we focus on the analysis of the receiver side since the receiver is under considerably more system strain than the sender.

3.2 Mathematical Model for Data Receiving Process

Linux 2.6 is a preemptive multi-processing kernel whose scheduling policy is priority-based and is explicitly in favor of I/O bound processes in order to provide a fast process response time (interactive processes are I/O bound). Processes are initially assigned with static priorities, which can be modified dynamically by the scheduler to fulfill scheduling objectives. The Linux scheduler calculates a dynamic priority through the static priority and interactivity of the process. A process with a higher interactivity is assigned with a higher dynamic priority

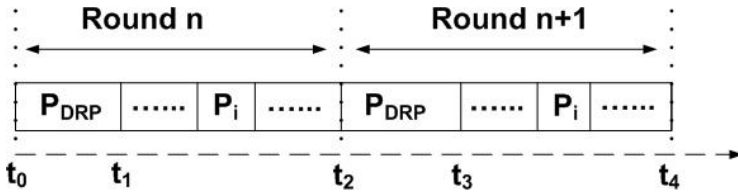


Fig. 3. Data receiving process running model (P_{DRP} representing the data receiving process)

and hence runs more frequently. On the contrary, CPU bound processes receive a lower dynamic priority. The timeslice of a process is determined by its dynamic priority per round of execution. Thus, important processes are assigned a longer timeslice that enables these processes to run longer. The old Linux CPU scheduler recalculates each task’s timeslice using an $O(n)$ algorithm implemented as a loop over each task; while the newer Linux scheduler maintains two priority arrays, an active array and an expired array, with $O(1)$ complexity for priority updating. Processes move from the active array to the expired array when they exhaust their timeslices. Recalculating all timeslices is just to switch the active and expired arrays [15].

Based on the above analysis, the running behavior of the data receiving process is shown in Fig. 3. Let t_{DRP} and t_{EXP} be the CPU time and the expired time assigned to the data receiving process, respectively, and t_{TOT} be the total CPU time assigned to all the running processes. We have:

$$t_{DRP} = timeslice(P_{DRP}). \tag{1}$$

$$t_{TOT} = timeslice(P_{DRP}) + \sum_{i=1}^n timeslice(P_i), \quad P_i \neq P_{DRP}. \tag{2}$$

The expired time for the data receiving process is:

$$t_{EXP} = t_{TOT} - t_{DRP}. \tag{3}$$

From Eqs. 1, 2 and 3, we know that the running time of the data receiving process is contingent on its own priority and the system load, which includes all interrupt-related processing and handling as well as the load of concurrent processes. Note that interrupt handling has the highest priority and is always scheduled to run before other tasks. Hence, a system with a high interrupt rate is not able to respond to the data receiving process immediately, resulting in a decreased data receiving rate. In an extreme case where the system is completely occupied for handling interrupts, the data receiving process could be temporarily suspended, resulting in significant packet losses in the socket receive buffer. Similarly, a system heavily loaded with concurrent processes could not guarantee enough CPU cycles for the data receiving process because processes with higher priorities

may starve the data receiving process. To increase the data receiving rate, one needs to either increase the data receiving process' priority or reduce the system load. However, reducing the system load does not seem to be a viable solution since the data receiving process typically runs with other concurrent resource-intensive workloads in a shared computing environment.

We denote the packet processing rate through the kernel protocol stack as λ and the effective service rate of the data receiving process as μ in the unit of bits per second (bps). Therefore, $\frac{1}{\mu}$ is the time the receiving process takes to copy an incoming packet from the kernel socket receive buffer into the application buffer. We wish to determine an appropriate size of the UDP's receiving buffer to match the kernel packet processing rate λ with the data receiving rate μ . Let t be the time in seconds and m be the UDP buffer size in bytes. The time to deplete m when the packet receiving process runs out of its time slice is given by:

$$t = \frac{m}{\lambda}. \quad (4)$$

On the other hand, the time to deplete m when the CPU time is available to process the arriving packets is given by:

$$t = \frac{m}{\lambda - \mu}. \quad (5)$$

At time t , the kernel socket receive buffer is not able to accept any new packets and thus will have to drop them. The depleted UDP buffer results in the drop of the UDP datagrams received by the kernel.

3.3 Predicting Bottleneck Processing Rate at the Receiver

We collected source rates, goodputs, loss rates and retransmission rates over the USN-ESnet hybrid channel using a UDP-based transport profile generator [18], as shown in Fig. 4. These profiles illustrate how the destination acknowledgement interval together with the source rate affects the transport performance over dedicated channels. We observed that the peak goodput is achieved with low loss and low retransmission rates, which inspires us to derive the desired bottleneck rate.

Let T_{ts} be the timeslice of the data receiving process in one round and T_{tp} be the average time required for copying one packet from the socket receive buffer to the application buffer. The average processing rate $\bar{\mu}$ in this round can be calculated as:

$$\bar{\mu} = \frac{T_{ts}}{T_{tp} \cdot t_{TOT}}. \quad (6)$$

We consider two cases. (i) If $\lambda > \bar{\mu}$, the socket receive buffer will become full after time t , as shown in Eqs. 4 and 5. In this case, the data receiving process is not able to consume all the packets arriving from the network, resulting in packet loss in the socket receive buffer. At high data rates, generating and sending packets retransmission requests at the receiver consume CPU time and may significantly interfere with the data receiving process. (ii) If $\lambda < \bar{\mu}$, the data

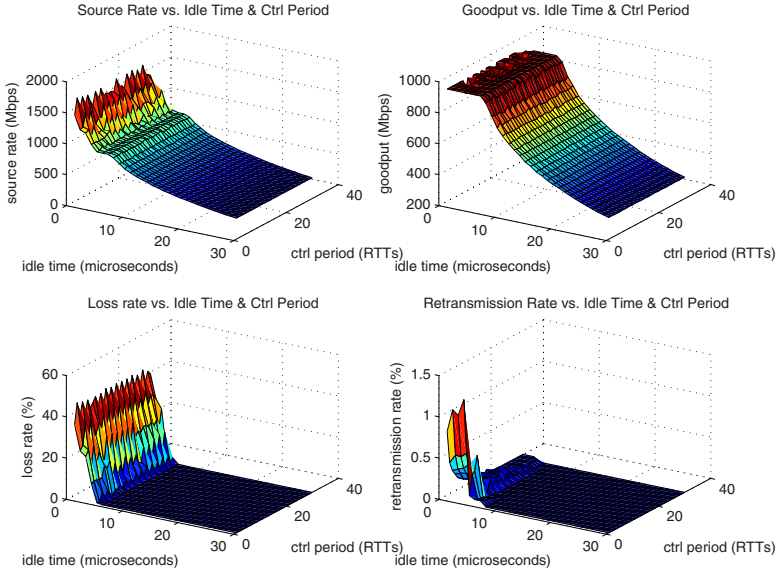


Fig. 4. Goodput, loss and re-transmission profiles of PLUT over 9900 mile 1Gbps USN-ESnet hybrid connection

receiving process has sufficient CPU cycles to consume the packets but there are no enough packets in the socket receiving buffer. In this case, the socket receiving buffer could become empty and there are still idle CPU cycles, both of which are a waste of system resources. The transport profiles show that the receiver cannot achieve the peak goodput in either case. So, $\bar{\mu}$ is the corresponding bottleneck processing rate for achieving peak goodput on the receiver side.

We know that Linux 2.6 is a preemptive multi-processing kernel whose scheduling policy is priority-based and is explicitly in favor of I/O bound processes in order to provide a fast process response time (interactive processes are I/O bound). The timeslice of a process is determined by its dynamic priority per round of execution. Thus, important processes are assigned a longer timeslice that enables these processes to run longer. So in order to get a longer timeslice to increase the value of $\bar{\mu}$, the data receiving process should be given a high priority.

In practice, we can sample μ at an carefully selected interval Δ . We denote such a sequence of μ samples as $\langle \mu_T \rangle = \dots \mu_{T-1} \mu_T \mu_{T+1} \dots$. If μ_{T+k} is known for $k > 0$, we could predict μ_{T+k+1} in some way. We define the following notations to facilitate the description of the prediction strategy:

- μ_T : the service rate of the data receiving process at the T -th measurement.
- μ_{T+1} : the prediction service rate for the $(T + 1)$ -th measurement.
- N : the number of historical data points used for the prediction of μ_{T+1} .

We measure the prediction quality by the *mean squared error*, which is the average of the square of the difference between predicted values and actual values.

We treat the sequence of periodic samples of μ as a linear time series. We employ the autoregressive (AR) model [3,6], a common approach for modeling univariate time series:

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + A_t, \tag{7}$$

where X_t is the time series, A_t is the white noise, and δ is a constant. The value of p is called the order of the AR model. After measuring μ_t , we can predict the value of μ_{t+1} at time $(t + 1)$ using the AR model.

4 Rate Control for Data Sender

Based on the performance model, the receiver sends the predicted bottleneck rate back to the sender periodically. If the sender just simply fix data sending rate at the bottleneck processing rate, it will not yield the highest goodput at the receiver which in turn involves accounting at some level for both network and host dynamics. Let $r_S(t)$ be the rate at which packets are sent and let $l(t)$ be the fraction of them that are lost before being read by the receiver, and hence have to be retransmitted. Let $x(t)$ be the fraction of $r_S(t)$ that corresponds to retransmitted packets. Thus the flow $r_S(t)$ is composed of two streams of rates $g_S(t)$ and $x(t)r_S(t)$ corresponding to packets sent for the first time and retransmissions, respectively. In general the data processing rate $\mu_R(t)$ at the receiver depends on $r_S(t)$, $l(t)$ and $x(t)$. The effect of randomness necessitates the utilization of stochastic approximation methods, which has a non-trivial effect on the underlying transport method: the step sizes used in parameter adaptation must be appropriately varied as per conditions such as in classical Robbins-Monro case [13]. To take into account the random effects, we define *processing-rate regression* as

$$G_R(r) = E [\hat{\mu}_R(t) | r_S(t) = r]. \tag{8}$$

Similarly, we have *loss-fraction* and *retransmission-fraction regressions* defined as

$$L(r) = E [\hat{l}(t) | r_S(t) = r] \quad \text{and} \quad X(r) = E [\hat{x}(t) | r_S(t) = r]. \tag{9}$$

Let μ^* be the attainable bottleneck processing rate at the receiver over a given dedicated connection. The objective of APPTC control is to stabilize $r(\cdot)$ at a suitable rate r^* , such that:

$$G_R(r^*) = \mu^* = r^*[1 - X(r^*)], \tag{10}$$

which ensures that peak throughput is attained at low loss rate.

At time step k , for the measured source rate $\hat{r}_S(k)$, measured processing rate $\hat{\mu}_R(k)$, and measured retransmission rate $\hat{x}(k)$, the equation $\hat{r}(k) = \hat{\mu}(k)/[1 - \hat{x}(k)]$ is only approximately satisfied. For $\hat{r}_S(k) = a(k) \cdot \mu^*(k)$ and $\hat{\mu}_R(k) = \alpha \cdot \mu^*(k)$, the coefficient function are typically $a(k) \geq 1$ and $\alpha(k) \leq 1$. Thus there are two possible estimates of $\mu^*(k)$ based on $\hat{r}_S(k)$ and $\hat{\mu}_R(k)$, which yield

two different values. We consider the following general form that combines these two estimates:

$$\hat{\mu}^*(k) = [\hat{r}_S(k)(1 - \hat{x}(k))]^\beta \hat{\mu}_R(k)^{1-\beta}, \quad 0 \leq \beta \leq 1, \tag{11}$$

where β is determined by host and link properties. Typically, $\hat{r}_S(k)$ and $\hat{x}(k)$ are more stable compared to $\hat{\mu}_R(k)$ since the former are not subject to connection-level variations. For the specific case where $\alpha(k) = 1/a(k)$, we have $\hat{\mu}^*(k) = \sqrt{\hat{r}_S(k)\hat{g}_R(k)}$. To account for randomness in measurements and the effects of delay and its variation of sending rate $\hat{r}_S(k)$ on processing rate measurement $\hat{\mu}_R(k)$, we apply a dynamic version of Robbins-Monro method [13] to adjust the source rate to achieve the target bottleneck processing rate $\mu^*(k)$ at the receiver:

$$\hat{r}_S(k + 1) = \hat{r}_S(k) - \rho_k[\hat{\mu}_R(k) - \hat{\mu}^*(k)], \tag{12}$$

where the time step adjustment coefficient is given by $\rho_k = b/k^\gamma$ for $0.5 < \gamma < 1.0$ and $b > 0$, a suitably chosen constant. The sending rate will increase if the measured processing rate $\hat{\lambda}_R(k)$ is less than the estimated maximum attainable processing rate $\hat{\mu}^*(k)$ at low sending rates; while in the source rate control zone approaching the peak processing rate, the processing rate measurement may exceed the maximum processing rate estimate due to increased retransmission rate, causing the sender to back off.

The step sizes satisfy the Robbins-Monro property namely, $\sum_{k=1}^\infty \rho_k = \infty$ and $\sum_{k=1}^\infty \rho_k^2 < \infty$. We assume that the errors satisfy the following martingale property for $\hat{r}_S(k) = r$:

$$E[\hat{g}(k) - \hat{g}^*(k) | \hat{r}_S(k) = r] = G_R(r) - [r(1 - X(r))]^\beta G_R(r)^{1-\beta},$$

which essentially assumes that the errors are not correlated across the time steps other than through $\hat{r}(\cdot)$. Then the limit behavior of Eq. 12 is specified by the Ordinary Differential Equation (ODE) (Chapter 5, [13]):

$$\frac{d\hat{r}}{dt} = E[\hat{\mu}^*(k) - \hat{\mu}_R(k)] = E[\hat{\mu}^*] - G_R(\hat{r}).$$

Under low loss condition, we approximate

$$E[\hat{\mu}^*] = [\hat{r}(1 - X(\hat{r}))]^\beta G_R(\hat{r})^{1-\beta}.$$

Then under the conditions (A.1), (A.3-4), the solution to ODE is given by the stationary point corresponding to

$$G_R(\hat{r}) \left[1 - \left(\frac{\hat{r}[1 - X(\hat{r})]}{G_R(\hat{r})} \right)^\beta \right] = 0,$$

which in turn corresponds to $G_R(\hat{r}) = \hat{r}[1 - X(\hat{x})] = \mu^*$. Thus the limit behavior of this algorithm is to stabilize at sending rate $\hat{R}_S(k) \rightarrow \hat{r}$ such that $\hat{\mu}_R(k) \rightarrow \mu^*$

Table 1. Goodput performance comparison (Mbps) without concurrent workloads

10 cases without load	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Std. Dev.
PAPTC	878	860	867	863	860	864	872	860	869	868	5.915
UDT	835	861	836	860	826	857	848	848	842	832	12.296
Tsunami	669	662	679	687	667	667	664	666	696	680	11.275

as $k \rightarrow \infty$. Alternatively, the required stability property can be derived for this algorithm using the monotonic property of $G_R(\cdot)$ and $X(\cdot)$ to show this convergence result as in [4]. Thus, this step ensures that PAPTC probabilistically stabilizes at the bottleneck processing rate λ^* of the connection while ensuring the low loss rate. Informally, by maintaining $r_S(t) = \bar{\mu}$, we would achieve an average goodput of g^* , and an increase (decrease) in r^* results in an increase (decrease) in $M(r^*)$.

5 Implementation and Experimental Results

The proposed transport protocol is implemented according to the architecture shown by Fig. 1, written mostly in C++ on Linux operating system.

5.1 Types of Acknowledgment

The proposed PAPTC protocol is implemented in C++ in Linux. We consider four different types of acknowledgment at the receiver: NXT (Next), RXM (Retransmission), TNT (Timeout Next), and TMO (Timeout Retransmission). For every normal ACK control period, if all datagrams received so far are in continuity, an “NXT” ACK is generated and sent to the sender; otherwise if there are lost datagrams (i.e. “holes” in the datagram checklist), the receiver compiles a list of lost datagram sequence numbers and sends them with a “RXM” ACK. If no datagram is received within a certain period of time, a timeout event is triggered where the receiver sends either a “TNT” ACK if all datagrams received so far are in continuity, or a “TMO” ACK enclosing the lost datagram sequence number list if there are “holes” in the datagram checklist. For all ACK types, the receiver measures the current bottleneck processing rate and sends it to the sender as part of the acknowledgment. On the sender side, for each incoming acknowledgment, we apply rate control as described in Section 4 using the bottleneck processing rate measurements enclosed in the acknowledgment.

5.2 Experimental Results

For performance comparison, we run PAPTC, UDT (version 4.4) and Tsunami on a local dedicated connection, which is provisioned by a back-to-back link between two Linux boxes with kernel 2.6.27. Each Linux box is equipped with a 1 Gigabit NIC, AMD Athlon(tm) 64X2 Dual Core Processor 5000+, 2 GBytes of RAM, and 900 GBytes of SCSI hard drive. A CPU-bound program named cpuburn is specifically designed and executed to emulate concurrent host background

Table 2. Goodput performance comparison (Mbps) with 2 concurrent cpuburn processes

10 cases with 2 cpuburn proc.	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Std. Dev.
PAPTC	816	802	813	801	807	818	808	815	807	806	5.889
UDT	716	718	734	717	716	717	718	718	726	741	10.601
Tsunami	670	676	676	669	679	675	661	668	639	671	11.549

Table 3. Goodput performance comparison (Mbps) with 4 concurrent cpuburn processes

10 cases with 4 cpuburn proc.	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Std. Dev.
PAPTC	655	654	643	646	644	682	635	656	665	633	14.622
UDT	613	623	615	626	610	623	629	620	618	622	5.934
Tsunami	622	625	621	622	628	626	623	625	626	621	2.424

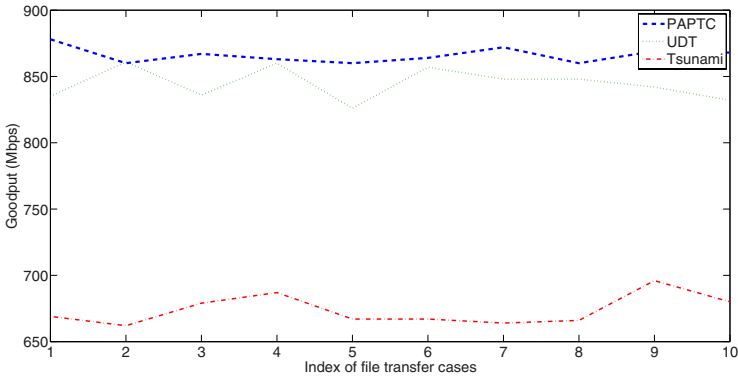


Fig. 5. Goodput performance comparison without concurrent load

workloads. We conduct three sets of transport experiments, in each of which, 10 files are transferred using three transport methods. In the first set of experiments, no cpuburn process is executed while in the other two sets of experiments, 2 and 4 concurrent cpuburn processes are launched, respectively. The goodput performance measurements and standard deviations for three transport methods are tabulated in Tables 1, 2, and 3, and their corresponding performance curves are plotted in Figs. 5, 6, and 7 for a better visual comparison. From these measurements, we observe that the amount of concurrent background workloads has a significant effect on the performance of each transport method. Tsunami is relatively insensitive to the change of concurrent workloads at a sacrifice of its goodput performance. Similar to PAPTC, UDT also adapts to the workload changes but adopts a somewhat more conservative rate control than PAPTC.

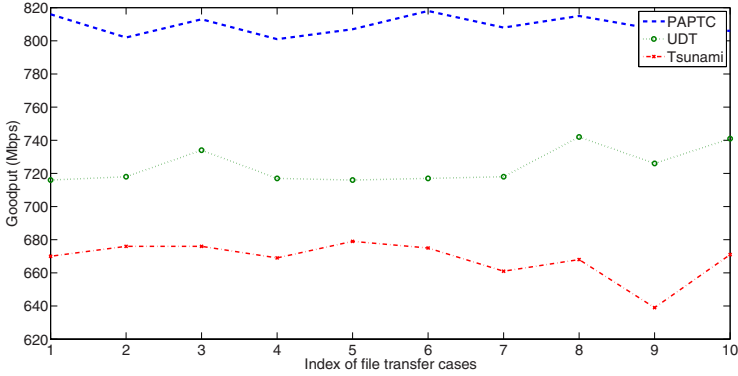


Fig. 6. Goodput performance comparison with 2 concurrent cpurn processes

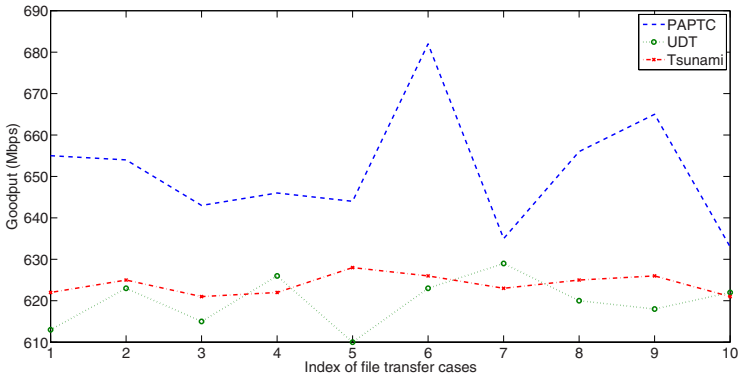


Fig. 7. Goodput performance comparison with 4 concurrent cpurn processes

In all the cases we studied, the proposed PAPTC protocol consistently achieves higher goodputs than the other two methods in comparison.

6 Conclusion

We developed PAPTC to support high-speed data transfers over dedicated channels. To account for the host dynamics and the random components in transport performance measurements, we designed control strategies based on performance modeling and stochastic approximations to achieve sustained high goodputs at a low packet loss. We implemented and tested PAPTC over a back-to-back connection and the experimental results illustrated its superior performance over existing methods.

Acknowledgments

This research is sponsored by National Science Foundation under Grant No. CNS-0721980 and Oak Ridge National Laboratory, U.S. Department of Energy, under Contract No. PO 4000056349 with University of Memphis.

References

1. DRAGON: Dynamic Resource Allocation via GMPLS Optical Networks, <http://dragon.maxgigapop.net>
2. Tsunami, <http://newsinfo.iu.edu/news/page/normal/588.html>
3. Beltran, M., Guzman, A.: A new cpu availability prediction model for time-shared systems. *IEEE Transaction* 2009 57, 865–875 (2009)
4. Benveniste, A., Metivier, M.: *Adaptive Algorithms and Stochastic Approximation*. Springer, New York (1990)
5. Brakmo, L., O'Malley, S., Peterson, L.: Tcp vegas: new techniques for congestion detection and avoidance. In: *SIGCOMM 1994 Conf. on Communications Architectures and Protocols*, London, United Kingdom, October 1994, pp. 24–35 (1994)
6. Dinda, P., OHallaron, D.: Host load prediction using linear models. *Cluster Computing* 3(4), 265–280 (2000)
7. Rio, M., et al.: A map of the networking code in linux kernel 2.4.20. Technical Report DataTAG-2004-1 (March 2004)
8. Floyd, S.: Highspeed tcp for large congestion windows, Internet Draft (February 2003)
9. Gu, Y., Hong, X., Mazzucco, M., Grossman, R.L.: SABUL: A high performance data transfer protocol. Submitted to *IEEE Communications Letters* (2004)
10. He, E., Leigh, J., Yu, O., DeFanti, T.: Reliable blast udp: predictable high performance bulk data transfer. In: *IEEE Int. Conf. on Cluster Computing*, Chicago, Illinois, September 23-26 (2002)
11. Katabi, D., Handley, M., Rohrs, C.: Internet congestion control for future high-bandwidth-delay product environments. In: *Proc. of ACM SIGCOMM 2002*, Pittsburgh, PA, August 19-21 (2002), <http://www.acm.org/sigcomm/sigcomm2002/papers/xcp.pdf>
12. Kelly, T.: Scalable tcp: Improving performance in highspeed wide area networks. In: *Workshop on Protocols for Fast Long-Distance Networks* (February 2003)
13. Kushner, H.J., Yin, C.G.: *Stochastic Approximation Algorithms and Applications*. Springer, New York (1997)
14. Kuzmanovic, A., Knightly, E., Cottrell, R.L.: Hstep-lp: A protocol for low-priority bulk data transfer in high-speed high-rtt networks. In: *The Second Int. Workshop on Protocols for Fast Long-Distance Networks* (February 2004)
15. Love, R.: *CPU Scheduler*. Sams (2003)
16. Low, S., Peterson, L., Wang, L.: Understanding vegas: a duality model. *J. of the ACM* 49(2), 207–235 (2002)
17. Rao, N., Wing, W., Carter, S., Wu, Q.: Ultrascience net: Network testbed for large-scale science applications. *IEEE Communications Magazine* 43(11), s12–s17 (2005), <http://www.csm.ornl.gov/ultranet>

18. Wu, Q., Rao, N.: Protocol for high-speed data transport over dedicated channels. In: Proc. of the 3rd Int. Workshop on Protocols for Fast Long-Distance Networks, February 3-4, pp. 155–162 (2005)
19. Zheng, X., Mudambi, A., Veeraraghavan, M.: Frtp: Fixed rate transport protocol – a modified version of sabul for end-to-end circuits. In: Proc. of Broadnets (2004)
20. Zheng, X., Veeraraghavan, M., Rao, N., Wu, Q., Zhu, M.: Cheetah: Circuit-switched high-speed end-to-end transport architecture testbed. *IEEE Communications Magazine* 43(11), s11–s17 (2005)

Probabilistic Network Loads with Dependencies and the Effect on Queue Sojourn Times

Matthias Ivers and Rolf Ernst

TU Braunschweig, Institute of Computer and Communication Network Engineering,
Hans-Sommer-Strasse 66, 38106 Braunschweig
ivers@ida.ing.tu-bs.de

Abstract. For the dimensioning of shared resources, the latency and utilization of the service is a vital design characteristic. The throughput and latency is as important for e.g. network streaming applications as in e.g. (small-scale) distributed embedded systems interacting with physical processes.

Calculating latencies of a system involves the analysis of the queue sojourn times. The analysis of queue sojourn time depends on the model of the load. While for fixed and known load, natural and deterministic worst-case models are a good choice, highly variable loads are more appropriately modeled in a stochastic fashion.

For the analysis of stochastic load models, the load is often assumed to be stochastic independent and time-invariant. Analysis of loads with auto-correlation or modelling of different streams that are correlated (or dependent in general) requires a highly tuned and specialized model to capture all effects.

In this work we apply a queuing sojourn time analysis of streams with stochastic load models with upper and lower bounds guaranteed under any stochastic dependency. The experimental results show how big the effect of dependencies really is and that stochastic load dependency is vital to the calculation of resource utilization and response times (or transmission delays). We propose the use of Fréchet bounds and probability boxes to allow real-time analysis of stochastic models with unknown dependencies.

1 Introduction

Safety critical systems rely on the timely processing of all signals and guarantee a reaction before a deadline, a minimum throughput or other performance characteristics. The duration of processing is as important as the correctness of the result under hard real-time requirements. The violation of a single deadline is considered fatal. To verify the correct operation, a real-time analysis that calculates upper bounds for the system response time has to be performed. In order to allow for tractable analysis, the components of the systems are abstracted. Deterministic real-time analysis characterizes the load by an upper bound of the processing time. In uniprocessor scheduling analysis this is called the worst-case execution time (WCET). Variable execution time due to input data or state

dependency or performance variation due to unpredictable systems (e.g. out-of-order processors and caches) are not captured. Thus deterministic analysis can lead to severe overestimations as the WCET can be much higher than the average execution time.

Stochastic uniprocessor performance analysis, which were pioneered in soft or firm real time systems, tackle this pessimism by modelling the execution time of tasks more precise: instead of a single worst case execution time, a task is characterized by a series of execution times together with a probability.

The precision of the load descriptions comes with a higher computational cost and, more important, further restrictions that are imposed on the analysed system: The mathematical foundation of the proposed stochastic analyses requires the explicit knowledge of every stochastic load dependency.

We want to *assess the effects of (uncaptured) dependencies between tasks*. The dependency information between distributions is of critical importance to the correctness of the results. For researchers in risk management the effect of dependency in stochastic models is well known. Initial work by Bernat et al. recasts methods presented for financial research into the realm of worst-case execution time analysis. Our work is based on the findings presented in [2]. We published parts of the presented work also in [9].

The presented work is based on scheduling analysis for uniprocessors. The central aspect of our work, the *dependency aware analysis of stochastic models* is not at all limited to the domain of uniprocessor scheduling. The central results can be transferred quite straightforward to other domains of stochastic analysis, as our model assumes a system with a single prioritized queue, a single (preemptive) server, a deterministic arrival model and a general load model *with possible stochastic dependencies*.

2 Related Work

Timing analysis of real-time systems traditionally characterizes tasks by their period and worst case execution time (WCET). The WCET of tasks is determined by different analytical methods. Methods like [15] decompose the analysis of tasks into an analysis of the runtime of code segments without (or more advanced with a limited number of) conditional branches and the synthesis of an upper bound WCET of all possible analyzed code segments that constitute the task under analysis.

These deterministic analysis yields a single value as WCET which is true under all circumstances. This WCET is then reused in a scheduling and system analysis to calculate end-to-end reponse times. Eventually end-to-end response times are compared against the application defined deadlines. Due to the high lever of abstraction, deterministic models, that characterize system load by a single WCET value, yield poor results for applications with highly variable system loads. E.g. multimedia streaming is a well-studied application domain where the consideration of the load variance improves analysis results considerably.

Specialized load models have been suggested for the analysis of multimedia streaming applications. These stochastic models try to capture a detailed presentation of the variance. Some models are so detailed, that they are fitted to the *content* and must be adapted, when video or audio content is modified. A survey [10] lists 19 different models proposed for VBR streaming applications. The models' statistical features can vary when exchanging application content. [17] identifies critical parameters in the dependency structure of streaming video application content that statistic models traditionally used fail to model. [12] concentrates on the autocorrelation of VBR video streams and gives examples for the effects on buffer sizing. These load models are specifically tailored to match the exact application *and* analysis method. Some of these models also demand lengthy computations or excessive memory. The proposed models do consider the intra stream dependencies of the load and also consider multiple levels of auto-correlation within the stream load. On the other hand, detailed analysis of the specific load is required and even more important, the computation with these models is highly resource consuming. We aim for a compositional system level analysis and the amount of specialization required for these models does not seem suit our goals, as some parts (or the environment) of the system might not be fully known during analysis (see below for effects of unknown statistical dependencies).

Concerning the stochastic scheduling analysis for systems, frameworks such as [1, 5, 14] execution time distributions. The distribution is due to changing input data requiring different processing, due to machine state (caches, branch predictors etc) or due to error and exception handling. The model is more precise, but the proposed analysis algorithms are only applicable if the tasks are stochastic independent [3]. The requirement of stochastic independence inhibits the applicability to a bigger class of systems.

Calculations with stochastic variables with unknown dependencies have been studied extensively and are a common tool in risk theory [4]. Bernat et al. use in their approach [3] only a single possible distribution which they assume as the worst case convolution, the comonotonous convolution. Later Bernat et al. [2] state that neither the assumption of independence, nor the comonotonicity are safe approximations and propose the use of the supremal convolution, which is an upper bound on all possible convolutions. Further they use copulas to separate the modelling of stochastic behaviour of a single task and the stochastic dependency between tasks.

In this work we focus on the cause and effect of stochastic dependencies in systems of multiple, concurrent, potentially interacting tasks. We integrate the safe calculations with stochastic variables under unknown dependencies with the stochastic scheduling analysis proposed by Diaz, Kim et al. Our methods give more insight into the system reponse not only by giving a reliable upper bound on the stochastic response times, but as well a lower bound. The difference between lower and upper bound gives direct feedback about the potential impact of dependencies.

3 Motivating Example

Consider the near-empty queue displayed in figure 1 which is part of a real-time systems with deadline requirements. The queue contains two jobs $j_{0,0}$ and $j_{0,1}$ that just arrived. We want to calculate the finish time $\mathcal{F}_{0,1}$ of the second job in the queue.

Both jobs belong to a task τ_0 with the observed execution time distribution $F_{\mathcal{C}'_0}$ shown in figure 2. The finish time $\mathcal{F}_{0,1}$ is easily found to be the sum of the individual execution times $\mathcal{F}_{0,1} = \mathcal{C}_{0,0} + \mathcal{C}_{0,1}$.

As the distribution of $\mathcal{C}_{0,0}$ and $\mathcal{C}_{0,1}$ is given, we can calculate response time distribution as the sum of the execution times *assuming mutual independence*.

If, however, the execution times are not mutually independent, the assumption can lead to wrong results. We will construct two corner-cases to explain the potential errors. For simplicity both examples rely on the fact that the jobs are colored and that the color changes the odds of the execution times.

We assume *red* takes 10 time units to process and *blue* takes 2 time units to process. Further we assume that the source feeding the queue acts in a color-keeping burst mode: it emits two events of the same color and pauses for a long time.

Thus the queue can only contain two *red* or two *blue* jobs, the resulting joint distribution is shown in the middle graph of figure 2. Comparing the left and the middle distributions, we can see that the initial assumption has a 0.25 underestimation of the probability that the joint processing takes at least 20 time units.

Assuming the same setup, but with the source constantly changing color and keeping the burst length of two jobs, we get a queue which can only contain one *red* and one *blue* job. This source will have a response time distribution given on the right of figure 2. The graph shows just a single possible execution time with probability 1.0; queues which take 2 time units on the first job, take 10 time units on the second job (and vice versa). This leads to a constant sum of 12 time units. Again we can see that the initial independency assumption underestimates the

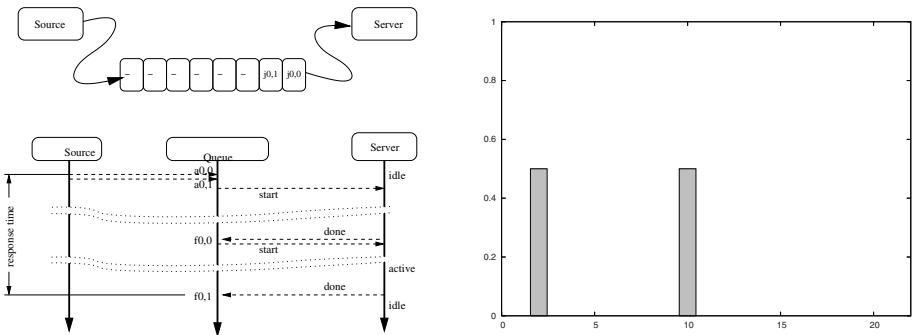


Fig. 1. Simple queue with a single task & the observed exec. time distribution

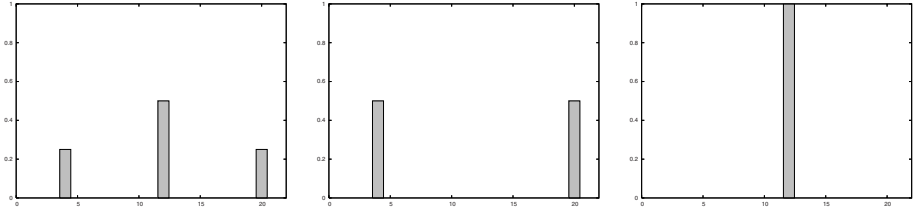


Fig. 2. $c_{0,0} + c_{0,1}$: (a) ind. (b) comonotonic (c) countermonotonic

probability: this time for the deadline of 12 time units. This difference translates directly into unsafe predictions. Assuming the system has a deadline requirement that $\mathcal{F}_{0,1}$ is below 20 time units in at least 70% of all cases ($P(\mathcal{F}_{0,1} < 20) \geq 0.70$). The first and the last graph of figure 2 do fulfill this requirement. The middle graph shows that the system would not meet the requirement $P(\mathcal{F}_{0,1} < 20) = 0.50$. The middle graph is, obviously, most pessimistic for the stated requirement. It is, however, not always most pessimistic, and thus cannot be used as *the worst case convolution*: Changing the deadline requirement to $P(\mathcal{F}_{0,1} < 10) \geq 0.25$ we can see that the first and the middle graph fulfill the requirement, but the third graph does predict system failure.

All these different scenarios are based on the same individual execution time distributions $F_{\mathcal{C}_0}$. This error in the analysis is due to the *uncaptured dependencies* between the random execution times. These dependencies have to be excluded or they have to be considered by the analysis to guarantee the correctness of the result.

4 Problem Statement

Given a set of tasks $S = \{\tau_1, \tau_2, ..\tau_n\}$. $\tau_i = (\mathcal{C}_i, d_i, M)$. Where \mathcal{C}_i is the execution time and d_i is the relative deadline of task τ_i and M is real number between 0 and 1. \mathcal{C}_i is a discrete random variable with probability mass function $f_{\mathcal{C}_i}(c) = P(\mathcal{C}_i = c)$ and cumulative probability function $F_{\mathcal{C}_i}(c) = P(\mathcal{C}_i \leq c)$. (Where $P(x)$ is the probability that in the specific system the event x is observed.) Furthermore a series of jobs $\tau_{i,j} = (\tau_i, a_{i,j})$ with task τ_i and an arrival time $a_{i,j}$ is given.

We assume a static priority preemptive scheduling. The tasks τ_i $i \in \mathbf{N}$ are w.l.o.g. ordered by priority. Jobs violating their deadlines are instantly killed.

For each job $\tau_{i,j}$ the response time is given by the random variable $\mathcal{R}_{i,j}$. Job $\tau_{i,j}$ is said to *fulfill its QoS requirement* (d_i, M) if $F_{\mathcal{R}_{i,j}}(d_i) \geq M$. We are seeking to find for all jobs $\tau_{i,j}$ bounds $(\overline{\mathcal{R}_{i,j}}, \underline{\mathcal{R}_{i,j}})$ to the distributions of the response time $\mathcal{R}_{i,j}$ fulfilling $\forall r \in \mathbf{N}. \overline{F_{\mathcal{R}_{i,j}}}(r) \geq F_{\mathcal{R}_{i,j}}(r) \geq \underline{F_{\mathcal{R}_{i,j}}}(r)$. These bounds should be valid *under arbitrary dependencies*.

5 Independent Jobs

First, we reproduce the results for the response time analysis of independent tasks. In the following we will extend the theory to handle unknown dependencies.

One main differentiator of the works is the definition how two random variables are added: Given two independent random variables \mathcal{X} and \mathcal{Y} the distribution of the sum $\mathcal{Z} = \mathcal{X} + \mathcal{Y}$ can be calculated as:

$$F_{\mathcal{Z}}(z) = \sum_{z=x+y} F_{\mathcal{X}}(x) * F_{\mathcal{Y}}(y)$$

This is the basic convolution that is assumed for the following works. We will later extend this addition to handle unknown dependencies.

[16] reinterpret the classical scheduling formula for probabilistic systems. For a job $\tau_{i,j} = (\tau_i, a_{i,j})$ and time interval $[a_{i,j}, a_{i,j} + t]$ let $\tau_{\chi_0}, \tau_{\chi_1}, \dots, \tau_{\chi_r}$ be the finite series of higher priority job arrivals from $a_{i,j}$ until $a_{i,j} + t$. τ_{χ_r} is the last job started before $a_{i,j} + t$. First an upper bound to the response time distribution at time t is defined:

$$\mathcal{R}_{i,j}^t = \mathcal{C}_i + \mathcal{C}_{\chi_0} + \mathcal{C}_{\chi_1} + \dots + \mathcal{C}_{\chi_r} \tag{1}$$

With \mathcal{C}_{χ_n} being the execution time distribution of τ_{χ_n} .

The probability for completion before the deadline is

$$\max\{F_{\mathcal{R}_{i,j}^t}(t) | t \in E\} \tag{2}$$

Where E is the set containing the $d_{i,j}$, the deadline of task $\tau_{i,j}$, and all arrival times of higher priority tasks before $d_{i,j}$. The authors remark the potential problems of the assumed independency in their approach.

Eq [1] is a straightforward extension of the deterministic case. It does not differentiate 'when' the interference happens. Consider a task with random execution time executing for 2 time units and then being interrupted (by a deterministic task) for 4 time units. The left part of figure [3] follows the assumption implicitly made in eq [1]. The right side of the figure shows how taking the offset between the preemption and the activation into account improves the response time analysis. In an analysis of deterministic tasks this 'interference-offset' does not have to be taken into account. This can be easily seen when the distribution of the task in fig [3] is exchanged by the distribution of a deterministic task with a single step of height 1.0 at $t \geq 3$ (see fig [4]).

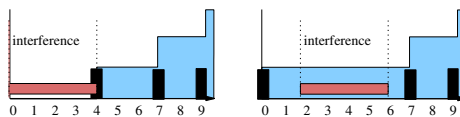


Fig. 3. Stochastic task interrupted - approximate and exact solution

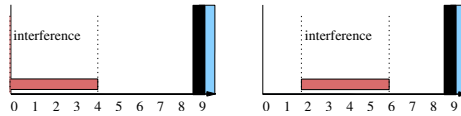


Fig. 4. Task with one possible execution time interrupted - interference offset does not matter

5 propose the operation “convolve from r ” that respects the condition of the interference: only instances that take longer than r are interfered.

The random variable \mathcal{Y} is added if and only if \mathcal{X} exceeds r . To achieve this, $F_{\mathcal{X}}$ is split into $F_{\mathcal{X}}^{[0,r]}$ and $F_{\mathcal{X}}^{(r,\infty)}$ where $x \in I \Rightarrow F_{\mathcal{X}}^I(x) = F_{\mathcal{X}}(x)$ otherwise $F_{\mathcal{X}}^I(x) = 0$.

The sum \mathcal{Z} of \mathcal{X} from r and \mathcal{Y} under independence (written $\mathcal{Z} = \mathcal{X} +_r \mathcal{Y}$) has the distribution:

$$F_{\mathcal{Z}}(z) = F_{\mathcal{X}}^{[0,r]}(z) + \sum_{z=x+y} F_{\mathcal{X}}^{(r,\infty)}(x) * F_{\mathcal{Y}}(y)$$

The operation $+_r$ is non-associative. We define the operation $+_r$ to be left-associative. For better readability parentheses are omitted.

The response time distribution is given by

$$\mathcal{R}_{i,j} = \mathcal{C}_i +_{\delta_{\mathcal{X}_0}} \mathcal{C}_{\mathcal{X}_0} +_{\delta_{\mathcal{X}_1}} \mathcal{C}_{\mathcal{X}_1} \dots +_{\delta_{\mathcal{X}_r}} \mathcal{C}_{\mathcal{X}_r} \tag{3}$$

$\delta_{\mathcal{X}_n} := a_{\mathcal{X}_n} - a_i$ is the arrival time difference between the analyzed job $\tau_{i,j}$ and $\tau_{\mathcal{X}_n}$.

Interference which occurs only under the condition that the execution took longer than a specific time now affects only that part of the distribution. The probability for completion within the deadline is thus $F_{\mathcal{R}_{i,j}}(d_i)$.

6 Dependencies in Execution Times

The previous analysis assumed for all random variables independency. The results are only valid if we can assure for any two jobs $\tau_{i,j}, \tau_{i',j'}$ in the system that the execution time distribution of one job does not change when another job has a certain execution time ($\mathcal{C}_{i',j'} = c$).

$$\forall (i, j) \neq (i', j'). P(\mathcal{C}_{i,j} = c) = P(\mathcal{C}_{i,j} = c | \mathcal{C}_{i',j'} = c') \tag{4}$$

The term dependency is also commonly used in the sense of “one task waiting for another task”. In this work “dependency” signifies a relationship between execution times only and not a precedence relationship.

Here *stochastic dependencies* describe all remaining influences on the response time distributions ($F_{\mathcal{R}}$) once the execution time distributions ($F_{\mathcal{C}}$) and activation times are fixed. We will depict two different types of dependency. First a dependency between the jobs of a single source and then a dependency between the jobs of different sources.

6.1 Sources with History

A *intra source dependency* is a dependency between two jobs of the same 'source'. This kind of dependency typically occurs in streams of jobs which have an inherent recurring regularity. An intra source dependency exists if

$$\exists n > 0. P(\mathcal{C}_{i,j} = x \mid \mathcal{C}_{i,j-n} = y) \neq P(\mathcal{C}_{i,j} = x) \quad (5)$$

An intra source dependency exists whenever the execution time of a job depends on the execution time of previous jobs.

An example for this scenario is the processing of MPEG streams. MPEG streams consist of a series of so-called I-, B-, and P-Frames. An I-Frame will typically be followed by a number of B- and P-Frames. Conversely the occurrence of two consecutive I-Frames is rare. As the different types of frames have very different transmission lengths and execution times associated with them, this qualifies as an intra source dependency.

In deterministic performance analysis for MPEG stream processors, these streams are modelled in a context-aware fashion. This means that the analysis is 'application-aware' and can handle this situation [8].

Another example for an intra source dependency, which is not due to the application model, are systems with caches: Consider two jobs executed back-to-back which require the same processing. The first invocation can suffer a cache miss, executes a long time while loading data into the cache. The second invocation, accessing the same data, will *not* suffer this cache miss and thus will execute faster. In this situation, the long execution of the first run will not be followed by another long execution.

6.2 Synchronized Sources

We speak of an *inter source dependency* if two or more sources change the load characteristic at the same time. An inter source dependency exists if there are two jobs $(\tau_{i,j}, \tau_{i',j'})$ with

$$P(\mathcal{C}_{i,j} = x \mid \mathcal{C}_{i',j'} = y) \neq P(\mathcal{C}_{i,j} = x) \text{ with } (i, j) \neq (i', j') \quad (6)$$

As an example consider variable bitrate video streams with isolated processing of audio and video. Streaming rates depend on the 'level of action' inside a stream. Calm scenes are probably accompanied by calm sounds and scenes with higher activity typically have high activity at both levels, audio and video. If one task is 'nearly idle', the other is probably as well. This is an inter source dependency.

Another example is a system with exclusive-or style load balancing between two tasks. It is not uncommon for two system functions to be designed in such a way that only one task is highly loaded at a time, while the other task (sharing the same processing resource) is nearly idle. This happens for example if in a multimedia stream processing different codecs are implemented in different tasks. As only one codec is active at a time, only this will generate high load. The unused codecs will generate no load.

6.3 Modeling of Dependency

A colored task $\tau_i = (\mathcal{C}_{k_0}, \mathcal{C}_{k_1}, \mathcal{C}_{k_2}, \dots, d_i)$ is a deadline d_i , a set of colors k_0, k_1, \dots and for each color a random response time \mathcal{C}_{k_0} . A colored job $\tau_{i,j} = (\tau_i, \alpha_{i,j}, k_{i,j})$ is a task with an activation time $\alpha_{i,j} \in \mathbf{N}$ and colored token $k_{i,j} \in \mathbf{K}$. The execution time $\mathcal{C}_{i,j}$ of task $\tau_{i,j}$ is chosen as determined by the color $k_{i,j}$. No other data than the color changes the odds of the random execution time $\mathcal{C}_{i,j}$. More formally

$$P(\mathcal{C}_{i,j} = c \mid k_{i,j} = k) = P(\mathcal{C}_{i,j} = c \mid k_{i,j} = k \vee \phi) \tag{7}$$

where ϕ is any formula which does *not* contain $\mathcal{C}_{i,j} = c'$. Using this model, we consider the job source as the generator of stochastic dependencies within the system. This model holds for a big class of systems and resembles multi-modal tasks in deterministic models.

6.4 Simulation of Effect

A simple example of a series of task activations with response time dependency will clarify the use of the model:

A single task t_1 with two possible execution times is periodically activated. The task has an arbitrary high deadline, and yet unprocessed tasks are stored in an unbounded queue. The execution time $c_{1,n}$ of job $t_{1,n}$ is determined by the following formula: $c_{1,n} = t_{1,off} + t_{1,mul} * x$. Where $t_{1,off}$ and $t_{1,mul}$ are non-negative task parameters and $x \in \{0, 1\}$ is a random variable (the color). Furthermore $t_{1,off} \geq t_{1,mul}$. The random variable x is determined by a source with one of the following strategies:

- independence: x changes from 0 to 1 (or vice versa) with a probability of 0.5
- positive dependence: x changes with a probability of 0.1
- negative dependence: x changes with a probability of 0.9

The tasks' execution time distribution $F_{\mathcal{C}_n}$ is identical for all dependencies. The resulting response-time has been plotted in figure 5. The figure shows the measured CDFs for the three scenarios. Comparing the three scenarios from left to right, you can identify that the end-to-end response time is dominated by the dependency of the tasks. The example demonstrates that a dependency has a significant effect on the response time distribution and should be taken into account.

The task's execution time distribution $F_{\mathcal{C}}$ is identical for all cases. The resulting response-time of the first 500 simulated tasks has been plotted in the bottom part of figure 5. The three lower graphs show the the execution time on the y-axis and the invocation number on the x-axis. The top part of the figure shows the measured CDFs for the three scenarios.

Comparing the three scenarios from left to right, it can be seen, that the backlog of the queues dominate the response time. The example demonstrates that an intra source dependency has a significant effect on the response time distribution and should be taken into account.

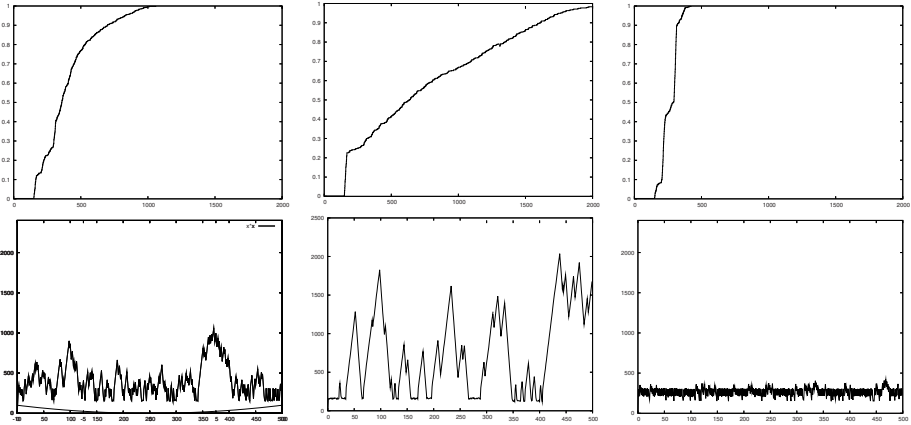


Fig. 5. Simulated response-time with (a) independent (b) + dependent and (c) - dependent tasks

7 Taking Dependencies into Account

Analyzing a job $\tau_{i,j}$, we have to consider all intra source dependencies for jobs of *higher priority* which can be activated multiple times before the deadline d_i (i.e. jobs of smaller period than τ_i). Additionally for all concurrently running higher priority jobs, we have to consider a potential inter source dependency for the sum of jobs of different priorities.

For this we have to use distribution functions that are able to represent uncertainty (i.e. distribution functions which can bound the probability for the value to be within a certain *interval*). To reason about interval bounds of stochastic variables, we will introduce so-called probability boxes.

7.1 Probability Bounds and Probability Boxes

A probability box (p-box) [6] is the generalisation of interval arithmetic in the realm of distribution functions. Given a cumulative distribution $F_X : \mathbb{N} \rightarrow [0, 1]$, a p-box is a pair of functions $\underline{F}_X, \overline{F}_X : \mathbb{N} \rightarrow [0, 1]$ with

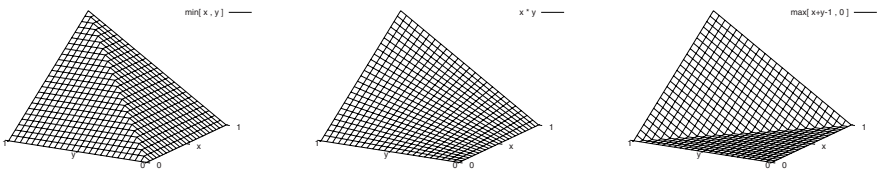


Fig. 6. M , H and W copulas

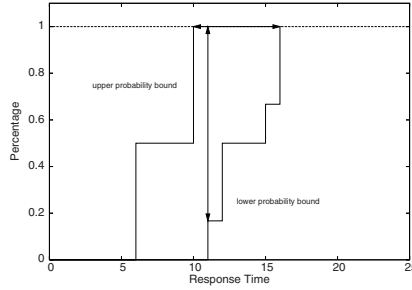


Fig. 7. A Probability Box

$$\underline{F}_{\mathcal{X}}(x) \leq F_{\mathcal{X}}(x) \leq \overline{F}_{\mathcal{X}}(x) \tag{8}$$

For any $F_{\mathcal{X}}$, $(\overline{F}_{\mathcal{X}}, \underline{F}_{\mathcal{X}}) := (F_{\mathcal{X}}, F_{\mathcal{X}})$ is a bounding p-box. Execution time distributions $F_{\mathcal{X}}$ map a time t_x to the probability that the execution time is less than or exactly t_x time units. Execution time p-boxes $(\overline{F}_{\mathcal{X}}, \underline{F}_{\mathcal{X}})$ map a time t_x to a minimum and maximum probability that the execution time is less than or exactly t_x time units (the interval for $t_x = 11$ is marked by the vertical arrow in figure 7). Formally, a p-box satisfies

$$\underline{F}_{\mathcal{X}}(x) \leq P(\text{task finished before } t_x) \leq \overline{F}_{\mathcal{X}}(x) \tag{9}$$

Where $P(x)$ is the probability that in the specific system the event x is observed.

Probability boxes can also be used to read an execution time interval for a given probability. E.g. the horizontal arrow in figure 7 shows that the WCRT (the response time with an accumulated probability of 100%) ranges from 10 to 16 time units. These values say nothing about the *best-case* response time; the meaning is that there may exist a system with a *worst-case* response time of only 10 time units. The *best-case* response time is found in the graph at the points where the probability bounds leave the 0%-line. The *best-case* is guaranteed to be between 6 and 11 time units.

Probability bounds can be efficiently described with probability boxes. The remaining question is how the sum of stochastic variables should be safely calculated. [7, 13, 18] study arithmetic on stochastic variables with unknown dependencies.

Copulas are used to formalize dependencies between stochastic variables. Copulas model the relation between (typically available) marginal distributions and their joint distribution. i.e.: Given a two-dimensional distribution function $F_{\mathcal{F}\mathcal{G}}(x, y)$ with (one-dimensional) marginals $F_{\mathcal{F}}(x)$ and $F_{\mathcal{G}}(y)$. Then there exists a copula C such that

$$\mathcal{H}(x, y) = C(\mathcal{F}(x), \mathcal{G}(y)) \tag{10}$$

In our situation, 2 marginals $F_{\mathcal{F}}$ and $F_{\mathcal{G}}$ for different tasks/jobs are given. The unknown dependency is modeled only by C .

Obviously C is a function $C : [0, 1] \times [0, 1] \rightarrow [0, 1]$. Furthermore it has been proven that all copulas satisfy

- $C(a, 0) = C(0, a)$ and $C(a, 1) = C(1, a)$ for all $a \in [0, 1]$
- they are 2-increasing: i.e. $C(a_2, b_2) - C(a_1, b_2) - C(a_2, b_1) + C(a_1, b_1) \geq 0$ for all $a_1 \leq a_2, b_1 \leq b_2$.

Given these constraints, there exists a unique smallest and a unique largest copula. Namely $W(a, b) := \max(a + b - 1, 0)$ and $M(a, b) := \min(a, b)$. These copulas are handy as all copulas satisfy

$$W(a, b) \leq C(a, b) \leq M(a, b) \tag{11}$$

This observation lead to the Fréchet bounds, that give upper and lower bounds on the effect the dependency between marginals can have on the joint distribution:

$$\max(F_{\mathcal{F}}(x) + F_{\mathcal{G}}(y) - 1, 0) \leq F_{\mathcal{H}}(x, y) \leq \min(F_{\mathcal{F}}(x), F_{\mathcal{G}}(y)) \tag{12}$$

Another commonly used copula is $\Pi(x, y) := x * y$ which models stochastic independency of two random variables. M, W and Π are shown in figure 6.

Copulas and especially the W copulas give lower bounds for the probability $P(\mathcal{X} \leq x_1 \wedge \mathcal{X} \leq y_1)$ given the probability for $P(\mathcal{X} \leq x_1)$ and $P(\mathcal{X} \leq y_1)$. This is closely related to the sum of random variables, yet a little extension is necessary as we are not interested in the probability of $P(\mathcal{X} \leq x_1 \wedge \mathcal{X} \leq y_1)$, but instead we search for any given z_1 the probability $P(\mathcal{X} + \mathcal{Y} \leq z_1)$.

[18] gives probability bounds for the sum of two stochastic variables $\mathcal{Z} = \mathcal{C}_1 + \mathcal{C}_2$.

$$\overline{F_{\mathcal{Z}}}(t) = \inf_{c_1+c_2=t} \{W^d(F_{\mathcal{C}_1}(c_1), F_{\mathcal{C}_2}(c_2))\} \tag{13}$$

$$\underline{F_{\mathcal{Z}}}(t) = \sup_{c_1+c_2=t} \{W(F_{\mathcal{C}_1}(c_1), F_{\mathcal{C}_2}(c_2))\} \tag{14}$$

Where $W^d(x, y) := x + y - W(x, y) = \min(u + v, 1)$ is the dual of W . In order to consider sums of p-boxes (instead of distribution functions as above) the function is extended to p-boxes. The sum of two stochastic variables described by probability boxes $(\overline{F_{\mathcal{Z}}}, \underline{F_{\mathcal{Z}}}) = (\overline{F_{\mathcal{C}_1}}, \underline{F_{\mathcal{C}_1}}) + (\overline{F_{\mathcal{C}_2}}, \underline{F_{\mathcal{C}_2}})$ is calculated as follows:

$$\overline{F_{\mathcal{Z}}}(t) = \inf_{c_1+c_2=t} \{\min(\overline{F_{\mathcal{C}_1}}(c_1) + \overline{F_{\mathcal{C}_2}}(c_2), 1)\} \tag{15}$$

$$\underline{F_{\mathcal{Z}}}(t) = \sup_{c_1+c_2=t} \{\max(\underline{F_{\mathcal{C}_1}}(c_1) + \underline{F_{\mathcal{C}_2}}(c_2) - 1, 0)\} \tag{16}$$

The proposed operation for the addition of two random variables has been proven to be safe and furthermore pointwise best-possible [18]. That means for any bound tighter than $(\overline{F_{\mathcal{Z}}}, \underline{F_{\mathcal{Z}}})$, one can find a dependency between the underlying random processes, that would lead to a violation of these (tighter) bounds.

7.2 Adaptions to Scheduling Analysis

We will now integrate Fréchet bounds and probability boxes into the scheduling analysis formula using the “convolve from” operation. In order to do that, we basically have to lift the “convolve from” operation from simple distributions to probability boxes. I.e. the inputs of the convolve from operation are now probability boxes, and the output of the convolve from operation are probability boxes as well.

As this lifting from distributions to probability boxes is intriguingly conclusive, we show directly the “convolve from” operation using probability boxes and the Fréchet bounds:

The sum of $(\overline{F_X}, \underline{F_X})$ from r and $(\overline{F_Y}, \underline{F_Y})$ under any dependence is bounded by $(\overline{F_Z}, \underline{F_Z}) = (\overline{F_X}, \underline{F_X}) +_r (\overline{F_Y}, \underline{F_Y})$ is defined:

$$\overline{F_Z}(z) = \overline{F_X}^{[0,r]}(z) + \inf_{z=x+y} \{ \min(\overline{F_X}^{(r,\infty)}(x) + \overline{F_Y}(y), 1) \} \quad (17)$$

$$\underline{F_Z}(z) = \underline{F_X}^{[0,r]}(z) + \sup_{z=x+y} \{ \max(\underline{F_X}^{(r,\infty)}(x) + \underline{F_Y}(y) - 1, 0) \} \quad (18)$$

The operation $+_r$ is non-associative. We define the operation $+_r$ to be left-associative. For better readability parentheses are omitted.

The response time distribution is given by

$$(\overline{F_{\mathcal{R}_{i,j}}}, \underline{F_{\mathcal{R}_{i,j}}}) = F_{\mathcal{C}_i} +_{\delta_{x_0}} F_{\mathcal{C}_{x_0}} +_{\delta_{x_1}} F_{\mathcal{C}_{x_1}} \dots +_{\delta_{x_r}} F_{\mathcal{C}_{x_r}} \quad (19)$$

δ_{t_n} is the arrival time difference between the analyzed task and t_n .

The equation is well defined, as we can write $F_{\mathcal{C}_i}$ for the bounding p-box $(\overline{F_{\mathcal{C}_i}}, \underline{F_{\mathcal{C}_i}})$ with $\overline{F_{\mathcal{C}_i}} = \underline{F_{\mathcal{C}_i}} = F_{\mathcal{C}_i}$. Following our previous reasonings about the fréchet bounds we can see that

$$\forall t. \overline{F_{\mathcal{R}_{i,j}}}(t) \geq F_{\mathcal{R}_{i,j}}(t) \geq \underline{F_{\mathcal{R}_{i,j}}}(t) \quad (20)$$

Using this function for our scheduling analysis, we calculate safe and sharp bounds to the distributions. Additionally, we also get a notion of how much the missing dependency information is affecting the system, as we also calculate a lower bound.

8 Comparison

Kim, Diaz, et. al. present an scheduling analysis for priority-driven periodic real-time systems [11]. They give an example for the calculation of the response time distribution of a job based on a given interarrival pattern of different jobs and the job’s execution time given as a distribution. The jobs are assumed to be mutually independent in their execution time. The initial workload is assumed to be zero. Figure 9 presents a timeline with task activations and the ETPDFs of t_1 , t_2 and t_3 . Six graphs below the timeline show the remaining workload distribution at different times.

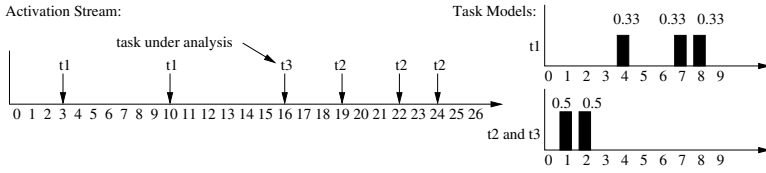


Fig. 8. Activation Diagram and Task Models

By time unit (tu) 3, task t_1 is activated: as the current workload is zero, the workload becomes exactly the task profile of t_1 . All graphs show results of Diaz’ original analysis and the proposed p-box based analysis. Diaz’ result is always within the p-box (the proposed method). In the first graph only a single line is visible, as both approaches are equal at 3 tu.

Until the next activation at 10 tu, no task is activated. To model the passing of time, the graph is *shrunked* i.e. shifted to the left and all probabilities for response times below zero are accumulated at 0. For details on *shrinking* see [11] or the examples below. Now, at 10tu, a second job starts. The task execution time distribution is added to the workload distribution. The result is presented in the second graph.

The third graph shows the workload distribution by the activation time of the task under analysis (16 tu). The workload distribution is composed of t_3 ’s own execution time distribution and the backlog distribution of the previous invocations.

The next three graphs show the workload distribution of the task t_3 at 16 time units considering the preemption at 16+3 (fourth graph), 16+6 (fifth graph) and 16+8 (sixth graph). The graphs are no longer shrunked, instead the activation time of the interrupting tasks is added starting from 3, 6 or 8 time units. The last graph shows the actual response time distribution of this invocation of t_3 (including all possible preemptions). This last graph is *the job response time*.

The big height of the p-box demonstrates how sensitive the system is to execution time dependencies. Remarkably the result assuming independence is quite far away from the ‘worst-case’ p-box bound. As the frechet bounds used to calculate the p-box are known to be sharp, we can assure that a wrongly assumed independency can practically lead to misleading optimistic results.

Two specific examples give insight how the dependencies might look like, that produce this big difference.

8.1 $x=3$ $y=1.0$ Example

Figure [10] shows the changing workload distribution on the left and arriving tasks on the right. All distributions are aggregated by colors ‘a’, ‘b’, ‘c’ showing the dependencies between the different task activations. As t_3 finishes within 3 timeunits, we do not give graphs for the succeeding invocations of t_2 .

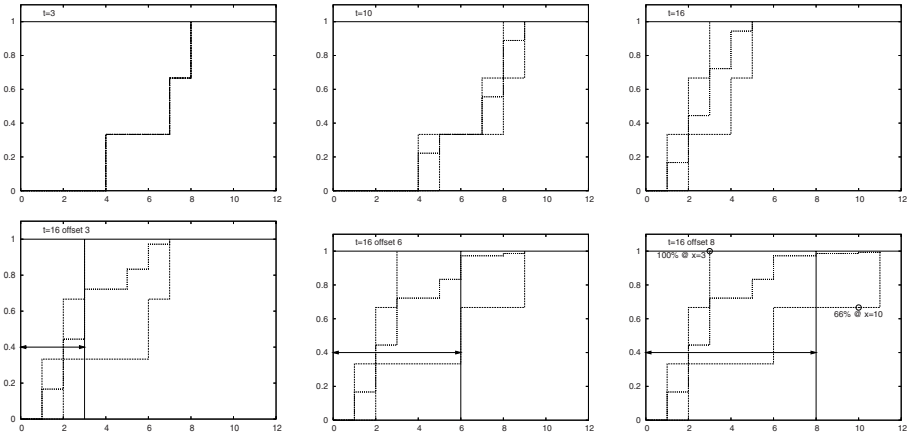


Fig. 9. First steps of Kim and Diaz' example (at time: 3, 10, 16, 16+3, 16+6, 16+8)

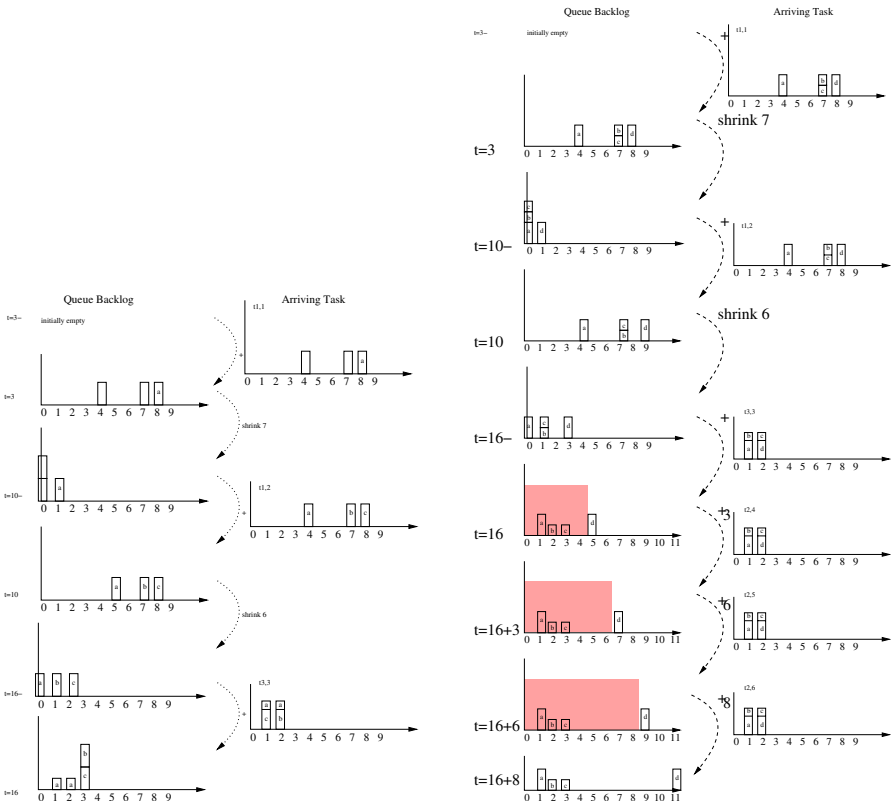


Fig. 10. Dependency pattern for $x=3, y=1.0$ (left) and for $x=10, y=0.66$ (right)

- Color 'a': if $t_{1,1}$ takes 8 time units, $t_{1,2}$ takes 4 time units. 15 tus after the activation of $t_{1,1}$, at the arrival time of $t_{3,1}$, no backlog is in the run-queue. $t_{3,1}$ is guaranteed to complete within 2 time units.
- N.B.: if $t_{1,1}$ takes less than 8 time units, the run-queue is empty at the arrival time of $t_{2,2}$. Thus $t_{1,1}$ only has to be considered if it executes 8 time units.
- Color 'b': if $t_{1,2}$ executes for 7 time units, $t_{3,3}$ executes for 2 time units. At the arrival time of $t_{3,3}$ the backlog will be one time unit leading to a total execution time of 3 time units.
- Color 'c': if $t_{1,2}$ executes for 8 time units, $t_{3,3}$ executes for 1 time units. At the arrival time of $t_{3,3}$ the backlog will be two time units leading to a total execution time of 3 time units.

The given dependency leads to $\mathcal{R}(3) = 1.0$. Under independence the guarantee is worse with $\mathcal{R}(3) \approx 0.722$.

8.2 $x=10$ $y=0.66$ Example

Figure 10 shows the queue distribution with the dependency to the new activations. For the distributions at $t = 16 + 3$ and beyond, only the case 'd' of the graph is affected, as this is the only case which is inside the manipulated part of the queue distribution ($+_n$ only affects the part of the distribution which is $> n$).

The given dependency leads to $\mathcal{R}(10) \approx 0.667$. Under independence the guarantee is far more optimistic with $\mathcal{R}(10) \approx 0.993$.

9 Conclusion

We demonstrated the effect of uncaptured dependencies in stochastic system analysis and have introduced probability boxes to describe uncertain probabilities instead of resorting to comonotonicity as a worst-case measure. Our examples demonstrate the magnitude of the effect and that neither the assumption of independence, nor the assumption of comonotonicity leads to safe estimations of the system behaviour.

Using our frechet-bound based 'convolve from' operation, we generate safe and sharp bounds for the analysis results. Using these methods, we can construct a compositional system analysis. Furthermore by the calculation of the upper and lower bound, we give a measure of the expected variation due to unknown dependencies.

References

1. Atlas, A.K., Bestavros, A.: Statistical rate monotonic scheduling, pp. 123–132. Boston University, Computer Science Department (1998)
2. Bernat, G., Burns, A., Newby, M.: Probabilistic timing analysis: An approach using copulas. *J. Embedded Comput.* 1(2), 179–194 (2005)

3. Bernat, G., Colin, A., Petters, S.M.: Wcet analysis of probabilistic hard real-time systems. In: Proceedings of the 23rd Real-Time Systems Symposium, RTSS 2002, pp. 279–288 (2002)
4. Chebyshev, P.: Sur les valeurs limites des integrales. *Journal de Mathematiques Pures Appliques* (1874)
5. Díaz, J.L., García, D.F., Kim, K., Lee, C.-G., Bello, L.L., López, J.M., Min, S.L., Mirabella, O.: Stochastic analysis of periodic real-time systems. In: RTSS 2002: Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS 2002), Washington, DC, USA, p. 289. IEEE Computer Society, Los Alamitos (2002)
6. Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D.S., Sentz, K.: Constructing probability boxes and dempster-shafer structures. Technical report, Sandia National Laboratories (2002)
7. Frank, M., Nelsen, R., Schweizer, B.: Best-possible bounds for the distribution of a sum- a problem of kolmogorov. *Prob. Theory Related Fields* (1987)
8. Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K., Ernst, R.: System level performance analysis - the symta/s approach. In: IEE Proceedings Computers and Digital Techniques (2005)
9. Ivers, M., Ernst, R.: Effect of Stochastic Load Dependencies on Queue Sojourn Times. In: Proceedings of 4th International Workshop on Real-Time Software (2009)
10. Izquierdo, M.R., Reeves, D.S.: A survey of statistical source models for variable bit-rate compressed video. *Multimedia Systems* 7, 199–213 (1999)
11. Kim, K., Diaz, J.L., Lopez, J.M., Lo Bello, L., Lee, C.-G., Min, S.L.: An exact stochastic analysis of priority-driven periodic real-time systems and its approximations. *IEEE Trans. Comput.* 54(11), 1460–1466 (2005)
12. Krunz, M.: The correlation structure for a class of scene-based video models and its impacts on the dimensioning of video buffers. *IEEE Trans. Multimedia* 2, 27–36 (2000)
13. Makarov, G.: Estimates for the distribution function of a sum of two random variables when the marginal distributions are fixed. *Theory Probab. Appli.* (1981)
14. Manolache, S.: Analysis and optimisation of real-time systems with stochastic behaviour. PhD thesis, Linköpings universitet – Institute of Technology (2005)
15. Staschulat, J., Ernst, R., Schulze, A., Wolf, F.: Context sensitive performance analysis of automotive applications. In: DATE 2005: Proceedings of the conference on Design, Automation and Test in Europe, Washington, DC, USA, pp. 165–170. IEEE Computer Society, Los Alamitos (2005)
16. Tia, T.-S., Deng, Z., Shankar, M., Storch, M., Sun, J., Wu, L.-C., Liu, J.-S.: Probabilistic performance guarantee for real-time tasks with varying computation times. In: Real-Time and Embedded Technology and Applications Symposium, p. 164. IEEE, Los Alamitos (1995)
17. Varatkar, G., Marculescu, R.: Traffic analysis for on-chip networks design of multimedia applications. In: DAC 2002: Proceedings of the 39th conference on Design automation, pp. 795–800. ACM, New York (2002)
18. Williamson, R.C., Downs, T.: Probabilistic arithmetic. i. numerical methods for calculating convolutions and dependency bounds. *Int. J. Approx. Reasoning* 4(2), 89–158 (1990)

Providing Performance Guarantees for Buffered Crossbar Switches without Speedup

Deng Pan, Zhenyu Yang, Kia Makki, and Niki Pissinou

Florida International University
Miami, FL 33199

{pand, yangz, makkik, pissinou}@fiu.edu

Abstract. Buffered crossbar switches are special crossbar switches with each crosspoint equipped with a small exclusive buffer. The crosspoint buffers decouple input ports and output ports, and simplify switch scheduling. In this paper, we propose a scheduling algorithm called Fair and Localized Asynchronous Packet Scheduling (FLAPS) for buffered crossbar switches, to provide tight performance guarantees. FLAPS needs no speedup for the crossbar and handles variable length packets without segmentation and reassembly (SAR). With FLAPS, each input port and output port independently make scheduling decisions and rely on only local queue statuses. We theoretically show that a crosspoint buffer size of $4L$ is sufficient for FLAPS to avoid buffer overflow, where L is the maximum packet length. In addition, we prove that FLAPS achieves strong stability, and provides bounded delay guarantees. Finally, we present simulation data to verify the analytical results.

Keywords: Buffered crossbar switches, performance guarantees, speedup, stability.

1 Introduction

Crossbar switches provide nonblocking capabilities, and overcome the bandwidth limitation of bus based switches [1]. They have long been used as high speed interconnects in various computing environments, such as Internet routers, computer clusters, and system-on-chip networks. Traditional crossbar switches have no buffers at the crosspoints of the crossbar switching fabric, and packets have to be directly transmitted from input ports to output ports. Such switches work in a synchronous time slot mode and handle only fixed length cells [2]. When variable length packets arrive, they need to be segmented into fixed length cells at input ports. The cells are then used as the scheduling units and transmitted to output ports, where they are reassembled into original packets and sent to the output lines. This process is called segmentation and reassembly (SAR) [3].

With the development of VLSI technology, it has been feasible to integrate on-chip memories to the crossbar [4] - [7]. Buffered crossbar switches are a special type of crossbar switches, which have a small exclusive buffer at each crosspoint,

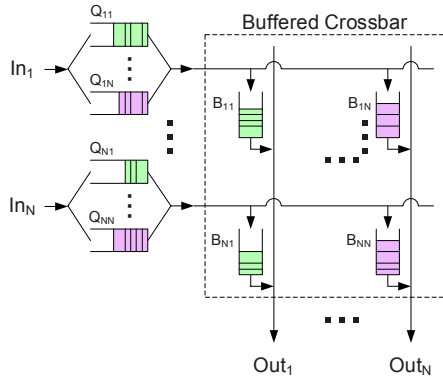


Fig. 1. Structure of buffered crossbar switches

as shown in Figure 1. Crosspoint buffers decouple input ports and output ports, and greatly simplify the scheduling process. Buffered crossbar switches can now directly handle variable length packets and work in an asynchronous mode. To be specific, input ports independently and periodically send packets of arbitrary length to their crosspoint buffers, from where output ports retrieve the packets one by one.

Compared with (fixed length) cell scheduling of unbuffered crossbar switches, (variable length) packet scheduling of buffered crossbar switches has some unique advantages. First, packet scheduling can better utilize available bandwidth and achieve higher throughput. For cell scheduling, when a packet is segmented into cells, its length may not be a multiple of the cell length, and padding bits have to be inserted to the last segment to reach the cell length. The padding bits do not contain useful information and waste bandwidth. In the worst case, if all packets have a slightly longer length than the cell length, each packet has to be segmented into two cells, and the switch can only achieve about a half of its maximum capacity [14]. Second, packet scheduling reduces packet latency, and helps achieve tight performance guarantees. Because there is no SAR, packets arriving at input ports can be immediately transmitted, and packets received at output ports can be immediately sent to the output lines. Third, no extra buffer space is necessary at input ports and output ports for SAR, which lowers hardware cost. In cell scheduling, an input port of an $N \times N$ switch may alternatively send segments of N packets to the N different output ports, and similarly an output port may receive segments from N input ports. Thus, NL buffer space is needed at each input port and output port for SAR, where L is the maximum packet length. Finally, cell scheduling is a special case of packet scheduling, or in other words, packet scheduling can also handle fixed length cells.

The speedup of a switch defines the ratio of the crossbar speed to the input or output port speed. To be specific, speedup of S means that the crossbar has S times bandwidth as that of the input port or output port. Obviously, with the same crossbar, the larger the speedup requirement is, the smaller the switch capacity is. The buffered crossbar switches considered in this paper do not need

speedup. Because the crossbar runs at the same speed as the output port, no buffer space is necessary at the output port. When a packet is transmitted to the output port, it will be immediately sent to the next hop via the output line.

There are a number of scheduling algorithms for buffered crossbar switches in the literature. Those algorithms can be classified into two categories: to provide performance guarantees [8] - [14] and to achieve high throughput [15] - [20]. The former requirement is stronger than the latter. In other words, an algorithm with tight performance guarantees usually delivers 100% throughput, but the reverse is not always true. Among the scheduling algorithms providing performance guarantees, some consider cell scheduling [8] - [13]. As discussed in the above, cell scheduling may waste bandwidth due to the padding bits in SAR. Others require speedup of two or more [9] - [14], which reduces the maximum capacity of the switch by half. In particular, two packet scheduling algorithms, Packet GVOQ (PGV) and Packet LOOFA (PLF), were proposed for buffered crossbar switches in [14], and their performance guarantees were analyzed. There are two main differences between the algorithms in [14] and our algorithm. First, PGV and PLF work by emulating push-in-first-out (PIFO) scheduling algorithms for output queued (OQ) switches, which means that they need to maintain the operation of the reference algorithms. Second, PGV and PLF require speedup of two for the crossbar. To the best of our knowledge, there have been no existing packet scheduling algorithms for buffered crossbar switches without speedup.

In this paper, we propose the Fair and Localized Asynchronous Packet Scheduling (FLAPS) algorithm for buffered crossbar switches without speedup. FLAPS allows input ports and output ports to make independent scheduling decisions based on only local information without data exchange. More specifically, an input port needs only the statuses of its input queues, and an output port needs only the states of its crosspoint buffers. FLAPS uses a time stamp based approach to schedule packets. We show that FLAPS has a crosspoint buffer size bound of $4L$, independent of the switch size. Furthermore, we prove that FLAPS achieves strong stability, and provides bounded performance guarantees. Finally, simulations are conducted to verify the analytical results and evaluate the performance of FLAPS.

The rest of the paper is organized as follows. In Section 2, we discuss the ideal fairness model that will be used. In Section 3, we present the FLAPS algorithm. In Section 4, we theoretically analyze the performance of FLAPS, and in Section 5, we present simulation data to verify the analytical results. In Section 6, we conclude the paper.

2 Preliminaries

In this section, we discuss the ideal fairness model that will be used in this paper. To effectively evaluate the fairness performance of a scheduling algorithm, it is necessary to have an ideal fairness model as the comparison reference. A fairness model for packet scheduling can be regarded to have two roles. The first role is to calculate the allocated bandwidth for traffic flows based on their

requested bandwidth. The second role is to schedule the packets of different flows to ensure that the actually received bandwidth of each flow is equal to its allocated bandwidth.

Generalized Processor Sharing (GPS) [21] is a widely used fairness model for packet scheduling. When GPS applies to a shared output link, it divides the link bandwidth into multiple logical transmission channels. Each flow has its own logical channel, and the channel bandwidth is proportional to the requested bandwidth of the flow. GPS views flows as fluids of continuous bits, and transmits the packets of a flow in its independent channel. As a result, each flow uses the same amount of bandwidth as that of its allocated bandwidth. To improve utilization, when a flow temporarily becomes empty, GPS will reallocate the leftover bandwidth of the empty flow to the remaining backlogged flows in proportion to their requested bandwidth.

As can be seen, when GPS serves a shared output link, it allocates available bandwidth, including leftover bandwidth, in the proportional manner. However, simple proportional bandwidth allocation is not proper for switch scheduling [22] [23]. The reason is that, while flows of a shared output link are constrained only by the link bandwidth, flows of a switch are subject to two bandwidth constraints: the available bandwidth at both the input port and output port of the flow. Naive bandwidth allocation at the output port may make the flows violate the bandwidth constraints at their input ports, and vice versa.

Fair bandwidth allocation for switches is an interesting problem, and there are algorithms [22] [23] in the literature to solve it. In this paper, we assume that bandwidth allocation has been calculated using such algorithms, and the scheduling algorithms just schedule packets to ensure the allocated bandwidth of each flow. Also, when a flow of the switch temporarily becomes empty, we do not assume that its allocated bandwidth is immediately reallocated. Instead, the bandwidth allocation algorithms will consider the leftover bandwidth in the next calculation. Bandwidth allocation is recalculated when requested bandwidth changes or existing backlogged flows become empty. Given the calculated bandwidth allocation, GPS can divide the bandwidth of a switch into logical channels, as shown in Figure 2. Each flow has an independent logical channel,

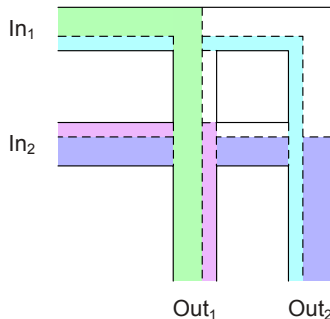


Fig. 2. GPS used for switch scheduling

and the channel bandwidth is equal to the allocated bandwidth of the flow. Thus, traffic of the flow can smoothly stream from the input port to the output port.

To sum up, we use GPS as the ideal fairness model only for its second role, i.e., given the allocated bandwidth, to compare the received bandwidth of a flow in our algorithm and in GPS.

3 Fair and Localized Asynchronous Packet Scheduling

In this section, we present the Fair and Localized Asynchronous Packet Scheduling (FLAPS) algorithm.

The switch structure that we consider is shown in Figure 1. N input ports and N output ports are connected by a buffered crossbar, which has no speedup. Denote the i^{th} input port as In_i and the j^{th} output port as Out_j . Use R to represent the available bandwidth of each input port and output port, and the crossbar also has bandwidth R . Each input port has a buffer organized as virtual output queues (VOQ) [24], i.e., there are N virtual queues to store the packets destined to the N different output ports. Denote the virtual queue at In_i for packets to Out_j as Q_{ij} . Each crosspoint has a small exclusive buffer. Denote the crosspoint buffer connecting In_i and Out_j as B_{ij} . Output ports have no buffers. After a packet arrives at the switch, it is first stored in the input queue, and waits to be sent to the crosspoint buffer. The packet will then be sent from the crosspoint buffer to the output port and immediately delivered to the output line. We say a packet arrives at or departs from a buffer when its last bit arrives at or departs from the buffer.

Define the traffic from In_i to Out_j to be a flow F_{ij} . Because we consider performance guarantees in this paper, each flow has explicit allocated bandwidth, and the objective of the scheduling algorithm is to ensure that each flow receives the same amount of bandwidth as that of its allocated bandwidth. Use $r_{ij}(t)$ to represent the allocated bandwidth of F_{ij} , which is a function of time t with discrete values in practice. Bandwidth allocation is calculated by the algorithms mentioned in Section 2. The calculated bandwidth allocation should be feasible, i.e., no over-subscription at any input port or output port

$$\forall i, \sum_{x=1}^N r_{ix}(t) \leq R, \text{ and } \forall j, \sum_{x=1}^N r_{xj}(t) \leq R \tag{1}$$

The feasibility requirement is only for bandwidth allocation. It is necessary because it is impossible to allocate more bandwidth than what is actually available. However, it does not mean that no temporary overload is allowed for an input port or output port. As will be seen in Section 4.2, we use the extended leaky bucket scheme for flow admission control. It allows any flow to be overloaded for an arbitrarily long period with an arbitrarily large but finite burst.

There are two types of scheduling in the switch, which we call input scheduling and output scheduling. In input scheduling, an input port selects a packet from one of its input queues, and sends it to the corresponding crosspoint buffer.

In output scheduling, an output port selects a packet from one of its cross-point buffers, and sends it to the output line. When we need to differentiate the scheduling algorithms used for input scheduling and output scheduling, we use the notation “A-B”. A is the scheduler for input scheduling, and B is the scheduler for output scheduling. A and B could be either FLAPS or GPS. If we do not care the scheduler for output scheduling, we use a * mark for B. For example, GPS-GPS means that GPS is used for both input scheduling and output scheduling. In such a scenario, each flow has an independent channel, and its traffic moves smoothly from the input output to the output port without buffering in the middle, as shown in Figure 2.

Input scheduling and output scheduling of FLAPS rely on only local information, and are conducted in an asynchronous and distributed manner. To be specific, an input port needs only the statuses of the queues in its input buffer, and does not exchange information with any crosspoint buffer or output port. Similarly, an output port needs only the statuses of its crosspoint buffers.

We first describe the input scheduling of FLAPS, which uses time stamps as scheduling criteria. There are two types of time stamps. For easy representation, we denote the k^{th} arrived packet of F_{ij} as P_{ij}^k . The first time stamp for P_{ij}^k is called virtual input start time, denoted as \widehat{IS}_{ij}^k , which is the service start time of P_{ij}^k at the input port in GPS-*. The second time stamp is virtual input finish time, denoted as \widehat{IF}_{ij}^k , which is the service finish time of P_{ij}^k at the input port in GPS-*. In other words, if GPS is the scheduler for input scheduling, \widehat{IS}_{ij}^k and \widehat{IF}_{ij}^k are the time that the first bit and last bit of P_{ij}^k leave Q_{ij} , respectively. \widehat{IS}_{ij}^k can be calculated as follows

$$\widehat{IS}_{ij}^k = \max \left(IA_{ij}^k, \widehat{IF}_{ij}^{k-1} \right) \tag{2}$$

where IA_{ij}^k is the arrival time of P_{ij}^k at the input port. \widehat{IF}_{ij}^k satisfies the following relationship

$$\int_{\widehat{IS}_{ij}^k}^{\widehat{IF}_{ij}^k} r_{ij}(x) dx = L_{ij}^k \tag{3}$$

where L_{ij}^k is the length of P_{ij}^k . Because $r_{ij}(t)$ has only discrete values in practice, \widehat{IF}_{ij}^k can be easily calculated. For example, if $r_{ij}(t)$ is a constant r_{ij} during $[\widehat{IS}_{ij}^k, \widehat{IF}_{ij}^k]$, \widehat{IF}_{ij}^k can be calculated as

$$\widehat{IF}_{ij}^k = \widehat{IS}_{ij}^k + \frac{L_{ij}^k}{r_{ij}} \tag{4}$$

In the first step of input scheduling of FLAPS, In_i identifies eligible packets. A packet P_{ij}^k is eligible for input scheduling if its virtual input start time \widehat{IS}_{ij}^k is

Table 1. Input Scheduling of FLAPS

```

for  $In_i$  do {
  while true do {
    if there are packets in local input queues with virtual input
      start time smaller than or equal to current system time {
      select among such packets the one with the smallest
        virtual input finish time, say  $P_{ij}^k$ ;
      send  $P_{ij}^k$  to crosspoint buffer  $B_{ij}$ ;
      // system time progressing by  $\frac{P_{ij}^k}{R}$ 
    }
    else {
      wait until the next earliest virtual input start time;
    }
  }
}

```

smaller than or equal to the current system time t . In other words, a packet that has started transmission in GPS-* is eligible in FLAPS-*. If there exist eligible packets in the input buffer, In_i will select among such packets the one P_{ij}^k with the smallest virtual input finish time \widehat{IF}_{ij}^k , and send it to B_{ij} . If there are no eligible packets, In_i will wait until the next earliest virtual input start time of a packet. Note that when In_i is waiting for an eligible packet, if an empty input queue has a new incoming packet, whose virtual input start time is equal to its arrival time, In_i should immediately start transmitting this new packet. For easy reading, the pseudo code description for input scheduling of FLAPS is given in Table 1.

We denote the actual input start time and finish time of P_{ij}^k in FLAPS-* as IS_{ij}^k and IF_{ij}^k , which are the time that the first bit and the last bit of P_{ij}^k leave Q_{ij} in FLAPS-*, respectively. Apparently

$$IF_{ij}^k = IS_{ij}^k + \frac{L_{ij}^k}{R} \tag{5}$$

Output scheduling of FLAPS is similar to input scheduling. There are also several time stamps for P_{ij}^k . The virtual output start time \widehat{OS}_{ij}^k and virtual output finish time \widehat{OF}_{ij}^k are the time that the first bit and last bit of P_{ij}^k leave B_{ij} in FLAPS-GPS, respectively. In other words, after P_{ij}^k is delivered to B_{ij} by FLAPS, if GPS is the scheduler for output scheduling, P_{ij}^k will start transmission at \widehat{OS}_{ij}^k and finish at \widehat{OF}_{ij}^k . \widehat{OS}_{ij}^k is calculated as

$$\widehat{OS}_{ij}^k = \max \left(OA_{ij}^k, \widehat{OF}_{ij}^{k-1} \right) \tag{6}$$

Table 2. Output Scheduling of FLAPS

```

for  $Out_j$  do {
  while true do {
    if there are packets in local crosspoint buffers with virtual
      output start time smaller than or equal to current system time {
      select among such packets the one with the smallest
        virtual output finish time, say  $P_{ij}^k$ ;
      send  $P_{ij}^k$  to the output line;
      // system time progressing by  $\frac{P_{ij}^k}{R}$ 
    }
    else {
      wait until the next earliest virtual output start time;
    }
  }
}

```

where OA_{ij}^k is the arrival time of P_{ij}^k at B_{ij} in FLAPS-*, and is equal to IF_{ij}^k by neglecting the propagation delay. \widehat{OF}_{ij}^k satisfies the following relationship

$$\int_{\widehat{OS}_{ij}^k}^{\widehat{OF}_{ij}^k} r_{ij}(x)dx = L_{ij}^k \tag{7}$$

Similarly, in output scheduling of FLAPS, Out_j first identifies eligible packets, and a packet P_{ij}^k is eligible if its virtual output start time \widehat{OS}_{ij}^k is smaller than or equal to the current system time t . If there are eligible packets in the crosspoint buffers, Out_j retrieves the one P_{ij}^k with the smallest virtual output finish time \widehat{OF}_{ij}^k , and sends it to the output line. Otherwise, it waits until there is an eligible packet. The pseudo code description for output scheduling of FLAPS is given in Table 2.

Correspondingly, OS_{ij}^k and OF_{ij}^k are the actual output start and finish time of P_{ij}^k , which are the time that the first bit and the last bit of P_{ij}^k leave B_{ij} in FLAPS-FLAPS, respectively. It is obvious that

$$OF_{ij}^k = OS_{ij}^k + \frac{L_{ij}^k}{R} \tag{8}$$

4 Performance Analysis

In this section, we theoretically analyze the performance of FLAPS. We show that FLAPS has a bounded crosspoint buffer size, achieves strong stability, and provides tight delay guarantees.

4.1 Crosspoint Buffer Size Bound

To avoid overflow at crosspoint buffers, we would like to find the maximum number of bits buffered at any crosspoint.

Based on the description of the FLAPS algorithm, we have the following properties.

Property 1. *For any packet, its actual input start time in FLAPS-* is larger than or equal to its virtual input start time in GPS-*, i.e.,*

$$IS_{ij}^k \geq \widehat{IS}_{ij}^k \tag{9}$$

Property 2. *For any packet, its actual output start time in FLAPS-FLAPS is larger than or equal to its virtual output start time in FLAPS-GPS, i.e.,*

$$OS_{ij}^k \geq \widehat{OS}_{ij}^k \tag{10}$$

First, we define some notations for input scheduling. We say that Q_{ij} is backlogged at time t , if there exists k such that $\widehat{IS}_{ij}^k \leq t \leq \widehat{IF}_{ij}^k$. Intuitively, Q_{ij} is backlogged at t if Q_{ij} has buffered bits at t in GPS-*. Define $\hat{q}_{ij}(t)$ to represent the backlog status of Q_{ij} at t . $\hat{q}_{ij}(t) = 1$ or 0 means that Q_{ij} is backlogged or empty at t .

Use $toB_{ij}(t_1, t_2)$ and $\widehat{toB}_{ij}(t_1, t_2)$ to represent the numbers of bits transmitted by F_{ij} from In_i to B_{ij} during interval $[t_1, t_2]$ in FLAPS-* and GPS-*, respectively. Based on the definition of GPS, $\widehat{toB}_{ij}(t_1, t_2)$ can be calculated as

$$\widehat{toB}_{ij}(t_1, t_2) = \int_{t_1}^{t_2} r_{ij}(x)\hat{q}_{ij}(x)dx \tag{11}$$

Next, we define some corresponding notations for output scheduling. We say that B_{ij} is backlogged at time t , if there exists k such that $\widehat{OS}_{ij}^k \leq t \leq \widehat{OF}_{ij}^k$. Define $\hat{b}_{ij}(t)$ to represent the backlog status of B_{ij} at t . $\hat{b}_{ij}(t) = 1$ or 0 means that B_{ij} is backlogged or empty at t .

Use $toO_{ij}(t_1, t_2)$ and $\widehat{toO}_{ij}(t_1, t_2)$ to represent the numbers of bits transmitted by F_{ij} from B_{ij} to Out_j during interval $[t_1, t_2]$ in FLAPS-FLAPS and FLAPS-GPS, respectively. $\widehat{toO}_{ij}(t_1, t_2)$ can be calculated as

$$\widehat{toO}_{ij}(t_1, t_2) = \int_{t_1}^{t_2} r_{ij}(x)\hat{b}_{ij}(x)dx \tag{12}$$

The following lemma gives the relationship between the service time of a packet in FLAPS-* and GPS-*.

Lemma 1. *For any packet, its actual input start time in FLAPS-* is less than or equal to its virtual input finish time in GPS-*, i.e.,*

$$IS_{ij}^k \leq \widehat{IF}_{ij}^k \tag{13}$$

The proofs of Lemmas 1 to 4 are similar to the performance analysis of WF²Q in [25]. They are omitted in this paper due to space limitations.

Correspondingly, there is a lemma for output scheduling.

Lemma 2. *For any packet, its actual output start time in FLAPS-FLAPS is less than or equal to its virtual output finish time in FLAPS-GPS, i.e.,*

$$OS_{ij}^k \leq \widehat{OF}_{ij}^k \tag{14}$$

The next lemma compares $toB_{ij}(t_1, t_2)$ and $\widehat{toB}_{ij}(t_1, t_2)$.

Lemma 3. *During interval $[0, t]$, the difference between the number of bits sent from input port In_i to crosspoint buffer B_{ij} in FLAPS-* and GPS-* is greater than or equal to $-L$ and less than or equal to L , i.e.,*

$$-L \leq toB_{ij}(0, t) - \widehat{toB}_{ij}(0, t) \leq L \tag{15}$$

For output scheduling, there is a similar lemma as follows.

Lemma 4. *During interval $[0, t]$, the difference between the number of bits sent from crosspoint buffer B_{ij} to output port Out_j in FLAPS-FLAPS and FLAPS-GPS is greater than or equal to $-L$ and less than or equal to L , i.e.,*

$$-L \leq toO_{ij}(0, t) - \widehat{toO}_{ij}(0, t) \leq L \tag{16}$$

The next lemma compares the number of bits transmitted by the same flow in the input scheduling of GPS-* and the output scheduling of FLAPS-GPS.

Lemma 5. *During interval $[0, t]$, the number of bits transmitted by flow F_{ij} from input port In_i to crosspoint buffer B_{ij} in GPS-* is less than or equal to that from crosspoint buffer B_{ij} to output port Out_j in FLAPS-GPS plus $2L$, i.e.,*

$$\widehat{toB}_{ij}(0, t) \leq \widehat{toO}_{ij}(0, t) + 2L \tag{17}$$

Proof. Assume that B_{ij} in FLAPS-GPS is empty immediately before time s and is continuously backlogged during $[s, t]$. If B_{ij} is not backlogged at t , then $s = t$.

By Lemma 3, we have $toB_{ij}(0, s) \geq \widehat{toB}_{ij}(0, s) - L$. Because B_{ij} is empty before s and backlogged after s in FLAPS-GPS, all packets arriving at B_{ij} before s have been transmitted to Out_j , and a new packet arrives at B_{ij} at s . Thus

$$\begin{aligned} \widehat{toO}_{ij}(0, s) &\geq toB_{ij}(0, s) - L \\ &\geq \widehat{toB}_{ij}(0, s) - 2L \end{aligned} \tag{18}$$

On the other hand, because B_{ij} is continuously backlogged during $[s, t]$, $\hat{b}_{ij}(t)$ is equal to 1 in the interval. Therefore

$$\begin{aligned} \widehat{toO}_{ij}(s, t) &= \int_s^t r_{ij}(x)\hat{b}_{ij}(x)dx \\ &= \int_s^t r_{ij}(x)dx \\ &\geq \int_s^t r_{ij}(x)\hat{q}_{ij}(x)dx \\ &= \widehat{toB}_{ij}(s, t) \end{aligned} \tag{19}$$

Adding (18) and (19), we obtain

$$\widehat{toO}_{ij}(0, t) \geq \widehat{toB}_{ij}(0, t) - 2L \tag{20}$$

The following theorem gives the bound for the crosspoint buffer size.

Theorem 1. *In FLAPS-FLAPS, the maximum number of bits buffered at a crosspoint buffer is upper bounded by $4L$, i.e.,*

$$toB_{ij}(0, t) - toO_{ij}(0, t) \leq 4L \tag{21}$$

Proof. By Lemma 4,

$$toO_{ij}(0, t) + L \geq \widehat{toO}_{ij}(0, t) \tag{22}$$

By Lemma 5,

$$\widehat{toO}_{ij}(0, t) + 2L \geq \widehat{toB}_{ij}(0, t) \tag{23}$$

By Lemma 3,

$$\widehat{toB}_{ij}(0, t) + L \geq toB_{ij}(0, t) \tag{24}$$

Summing the above equations, we have proved the theorem.

4.2 Switch Stability

We have shown in the above that FLAPS has a bounded crosspoint buffer size. In this subsection, we show that the lengths of input virtual queues are finite, and thus FLAPS achieves strong stability.

As discussed in Section 3, because each flow is allocated a specific amount of bandwidth, it is necessary to have admission control for the flow to avoid over-subscription. The leaky bucket scheme [1] is a widely used traffic shaping scheme, and we will use it for admission control. In the classical definition of a leaky bucket, the flow rate is a constant, which we extend in this paper to be a variable. The reason is that the allocated bandwidth of a flow may change

after bandwidth allocation calculations. Use $toI_{ij}(t_1, t_2)$ to denote the number of incoming bits of F_{ij} during interval $[t_1, t_2]$. If F_{ij} is leaky bucket $(r_{ij}(t), \sigma_{ij})$ compliant, then during any interval $[t_1, t_2]$

$$toI_{ij}(t_1, t_2) \leq \int_{t_1}^{t_2} r_{ij}(x)dx + \sigma_{ij} \tag{25}$$

where σ_{ij} can be an arbitrary positive constant and is called the burst size of F_{ij} . Intuitively, during any time interval, F_{ij} can have σ_{ij} more incoming traffic than what it can transmit.

Use $Q_{ij}(t)$ to represent the queue occupancy of Q_{ij} at t , i.e. the number of bits buffered in Q_{ij} at t . Define $X(t) = (Q_{11}(t), \dots, Q_{ij}(t), \dots, Q_{NN}(t))$, and use $\|X\|$ to represent the Euclidean norm of vector $X = (x_1, x_2, \dots, x_n)$, i.e.

$$\|X\| = \sqrt{\sum_{i=1}^n x_i^2} \tag{26}$$

Following the definition in [26], we say that a system of queues is strongly stable if

$$\limsup_{n \rightarrow \infty} E[\|X(t)\|] < \infty \tag{27}$$

Note that strong stability implies 100% throughput [26].

Theorem 2. *When flows are leaky bucket complaint, FLAPS is strongly stable.*

Proof. Assume that flow F_{ij} is leaky bucket $(r_{ij}(t), \sigma_{ij})$ compliant. Also assume that Q_{ij} is empty immediately before s and continuously backlogged during $[s, t]$. This indicates that all packets of F_{ij} arriving at Q_{ij} before s have finished transmission by s in GPS-*, and the next packet has not arrived. Therefore

$$toI_{ij}(0, s) \leq \widehat{toB}_{ij}(0, s) + L \tag{28}$$

During $[s, t]$, Q_{ij} is continuously backlogged, and thus

$$\widehat{toB}_{ij}(s, t) = \int_s^t r_{ij}(x)\hat{q}_{ij}(x)dx = \int_s^t r_{ij}(x)dx \tag{29}$$

Because the arrival traffic is leaky bucket $(r_{ij}(t), \sigma_{ij})$ compliant, we have

$$toI_{ij}(s, t) \leq \int_s^t r_{ij}(x)dx + \sigma_{ij} \tag{30}$$

By (28), (29), and (30)

$$toI_{ij}(0, t) \leq \widehat{toB}_{ij}(0, t) + L + \sigma_{ij} \tag{31}$$

We know from Lemma 3 that $\widehat{toB}_{ij}(0, t) \leq toB_{ij}(0, t) + L$. Thus

$$\begin{aligned} toI_{ij}(0, t) &\leq \widehat{toB}_{ij}(0, t) + L + \sigma_{ij} \\ &\leq toB_{ij}(0, t) + 2L + \sigma_{ij} \end{aligned} \tag{32}$$

By the definition of $Q_{ij}(t)$, we can obtain

$$Q_{ij}(t) = toI_{ij}(0, t) - toB_{ij}(0, t) \leq 2L + \sigma_{ij} \tag{33}$$

Since both L and σ_{ij} are finite, we have

$$\|X(t)\| = \sqrt{\sum_{i=1}^N \sum_{j=1}^N Q_{ij}(t)^2} < \infty \tag{34}$$

4.3 Delay Guarantees

In this subsection, we show that FLAPS can provide bounded delay guarantees. For easy analysis, we assume that the allocated bandwidth $r_{ij}(t)$ of F_{ij} is a constant r_{ij} during interval

$$\left[\min \left(IS_{ij}^k, \widehat{IS}_{ij}^k \right), \max \left(OF_{ij}^k, \widehat{OF}_{ij}^k \right) \right].$$

Use \widetilde{OF}_{ij}^k to denote the departure time of P_{ij}^k in GPS-GPS. By neglecting the propagation delay, we have $\widetilde{OF}_{ij}^k = \widehat{IF}_{ij}^k$. Similarly, OF_{ij}^k is the departure time of packet P_{ij}^k in FLAPS-FLAPS, if the propagation delay is neglected.

Theorem 3. *The difference between the departure time of packet P_{ij}^k in FLAPS-FLAPS and GPS-GPS is greater than or equal to $-L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{2}{R} \right)$ and less than or equal to $L \left(\frac{3}{r_{ij}} + \frac{2}{R} \right)$, i.e.*

$$-L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{2}{R} \right) \leq OF_{ij}^k - \widetilde{OF}_{ij}^k \leq L \left(\frac{3}{r_{ij}} + \frac{2}{R} \right) \tag{35}$$

Proof. First, we prove $OF_{ij}^k - \widetilde{OF}_{ij}^k \geq -L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{2}{R} \right)$. It is obvious that

$$OF_{ij}^k \geq OA_{ij}^k + \frac{L_{ij}^k}{R} = IF_{ij}^k + \frac{L_{ij}^k}{R} \tag{36}$$

Based on Property **1**, we know $\widehat{IS}_{ij}^k \leq IS_{ij}^k$ or in other words $\widehat{IF}_{ij}^k - \frac{L_{ij}^k}{r_{ij}} \leq IF_{ij}^k - \frac{L_{ij}^k}{R}$, and thus we obtain

$$\begin{aligned} \widetilde{OF}_{ij}^k &= \widehat{IF}_{ij}^k \\ &\leq IF_{ij}^k + L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{1}{R} \right) \\ &\leq OF_{ij}^k + L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{2}{R} \right) \end{aligned} \tag{37}$$

Next, we prove $OF_{ij}^k - \widetilde{OF}_{ij}^k \leq L \left(\frac{3}{r_{ij}} + \frac{2}{R} \right)$. Based on Lemma **2**, we know $OS_{ij}^k \leq \widehat{OF}_{ij}^k$ and thus $OF_{ij}^k \leq \widehat{OF}_{ij}^k + \frac{L_{ij}^k}{R}$. By Lemma **5**, we have $\widehat{toB}_{ij}(0, t) - \widehat{toO}_{ij}(0, t) \leq$

$2L$, and by Lemma 3, we have $toB_{ij}(0, t) - \widehat{toB}_{ij}(0, t) \leq L$. Combining them, we obtain $toB_{ij}(0, t) - \widehat{toO}_{ij}(0, t) \leq 3L$. This indicates that, after P_{ij}^k arrives at B_{ij} , the maximum queue length at B_{ij} in FLAPS-GPS is $3L$. Because B_{ij} is served by GPS output scheduling with fixed allocated bandwidth r_{ij} in FLAPS-GPS, we have

$$\widehat{OF}_{ij}^k \leq OA_{ij}^k + \frac{3L}{r_{ij}} \leq IF_{ij}^k + \frac{3L}{r_{ij}} \quad (38)$$

By Lemma 1, $IS_{ij}^k \leq \widehat{IF}_{ij}^k$ and thus $IF_{ij}^k \leq \widehat{IF}_{ij}^k + \frac{L_{ij}^k}{R}$. Combing the above equations, we obtain

$$\begin{aligned} OF_{ij}^k &\leq \widehat{OF}_{ij}^k + \frac{L_{ij}^k}{R} \\ &\leq IF_{ij}^k + \frac{3L}{r_{ij}} + \frac{L_{ij}^k}{R} \\ &\leq \widehat{IF}_{ij}^k + \frac{3L}{r_{ij}} + \frac{2L_{ij}^k}{R} \\ &\leq \widetilde{OF}_{ij}^k + L \left(\frac{3}{r_{ij}} + \frac{2}{R} \right) \end{aligned} \quad (39)$$

5 Simulation Results

We have conducted simulations to verify the analytical results obtained in Section 4 and evaluate the performance of FLAPS.

In the simulations, we consider a 16×16 buffered crossbar switch without speedup. Each input port and output port has bandwidth of 1G bps. Since FLAPS can directly handle variable length packets, we set packet length to be uniformly distributed between 40 and 1500 bytes [27]. For bandwidth allocation, we use the same model as that in [15] and [17]. The allocated bandwidth $r_{ij}(t)$ of flow F_{ij} at time t is defined by an unbalanced probability w as follows

$$r_{ij}(t) = \begin{cases} R \left(w + \frac{1-w}{N} \right), & \text{if } i = j \\ R \frac{1-w}{N}, & \text{if } i \neq j \end{cases} \quad (40)$$

When $w = 0$, In_i has the same amount of allocated bandwidth at each output port. Otherwise, In_i has more allocated bandwidth at Out_i , which is called the hotspot destination. Arrival of a flow F_{ij} is constrained by a leaky bucket $(l * r_{ij}(t), \sigma_{ij})$, where l is the effective load. We set the burst size σ_{ij} of every flow to a fixed value of 10,000 bytes, and the burst may arrive at any time during a simulation run. We use two traffic patterns in the simulations. For traffic pattern one, each flow has fixed allocated bandwidth during a single simulation run. l is fixed to 1 and w is one of the 11 possible values from 0 to 1 with a step of 0.1. For traffic pattern two, a flow has variable allocated bandwidth. l is one of the 10 possible values from 0.1 to 1 with a step of 0.1, and for a specific l value, a random permutation of the 11 different w values is used. Each simulation run lasts for 10 seconds.

5.1 Crosspoint Buffer Size

Theorem 1 in Section 4.1 gives the bound of the crosspoint buffer size as $4L$. In this subsection, we look at the maximum and average crosspoint buffer occupancies in the simulations.

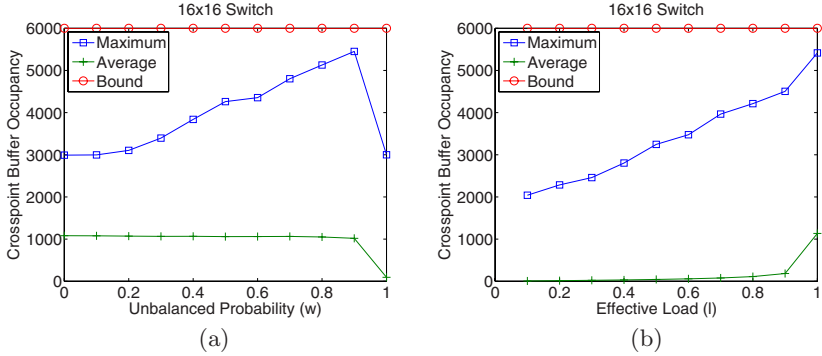


Fig. 3. Crosspoint buffer occupancy of FLAPS. (a) With different unbalanced probabilities. (b) With different loads.

Figure 3(a) shows the maximum and average crosspoint buffer occupancies under traffic pattern one. As can be seen, the maximum occupancy is always smaller than the theoretical bound. It grows as the unbalanced probability increases, but suddenly drops when the unbalanced probability becomes 1. This is because when the unbalanced probability is 1, all packets of In_i go to Out_i . Thus, there is no switching necessary, and the crosspoint buffer occupancy becomes smaller. For the average occupancy, it does not change significantly with different unbalanced probabilities, and drops when the unbalanced probability becomes 1 for the same reason. We can find that the average occupancy is more affected by the load than the unbalanced probability. Figure 3(b) shows the maximum and average crosspoint buffer occupancies under traffic pattern two. We can see that the maximum occupancy increases as the load increases, but does not exceed the theoretical bound. On the other hand, the average occupancy does not change much and is smaller than 200 bytes before the load increase to 1. This also confirms the previous observation that the average occupancy is determined by the load.

5.2 Throughput

Theorem 2 in Section 4.2 shows that FLAPS achieves strong stability, which implies 100% throughput. Next, we present the simulation data on throughput of FLAPS.

Figure 4(a) shows the throughput under traffic pattern one. We can see that the throughput for all unbalanced probabilities is greater than 99.99%, which

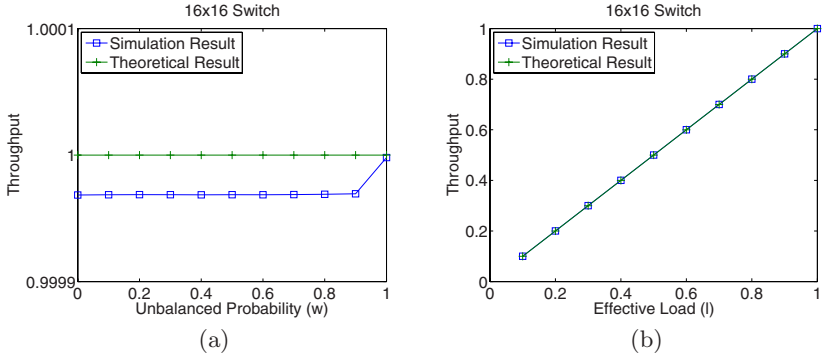


Fig. 4. Throughput of FLAPS. (a) With different unbalanced probabilities. (b) With different loads.

demonstrates that FLAPS practically achieves 100% throughput. Figure 4(b) shows the throughput under traffic pattern two. As can be seen, the throughput grows consistently with the effective load, and finally reaches 1.

5.3 Jitter

In this subsection, we present the simulation data on jitter, which is the difference between the packet departure time in FLAPS and GPS. Theorem 3 in Section 4.3 gives the lower bound and upper bound for the jitter of packet P_{ij}^k . Because Theorem 3 assumes fixed allocated bandwidth r_{ij} , we use only traffic pattern one for this part of simulations. Note that the lower bound value depends on the packet length L_{ij}^k . For easy plotting of the figure, we calculate the jitter lower bound for all packets of flow F_{ij} as follows

$$-L_{ij}^k \left(\frac{1}{r_{ij}} - \frac{2}{R} \right) \geq \begin{cases} -L \left(\frac{1}{r_{ij}} - \frac{2}{R} \right), & \text{if } r_{ij} \leq \frac{R}{2} \\ 0, & \text{if } r_{ij} > \frac{R}{2} \end{cases} \quad (41)$$

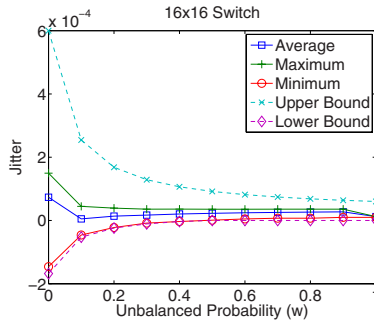


Fig. 5. Jitter of FLAPS with different unbalanced probabilities

Figure 5 shows the minimum, maximum, and average jitters of a representative flow F_{11} under traffic pattern one. We can see that the minimum jitter is almost coincident with but always greater than the lower bound, and the maximum jitter is always less than the upper bound. As the unbalanced probability increases, the minimum jitter increases and the maximum jitter decreases. For most of the time, the average jitter is very close to zero, indicating that FLAPS and GPS have similar average packet delay.

6 Conclusions

Buffered crossbar switches are special crossbar switches with crosspoint buffers. The introduction of crosspoint buffers greatly simplifies the scheduling process. In this paper, we have proposed the Fair and Localized Asynchronous Packet Scheduling (FLAPS) algorithm, which does not require speedup for the crossbar and can directly handle variable length packets without segmentation and reassembly (SAR). FLAPS uses a time stamp based approach for both input scheduling and output scheduling. We theoretically analyze the performance of FLAPS, and show that it has a crosspoint buffer size bound of $4L$, independent of the switch size. We also prove that it achieves strong stability, and provides bounded delay guarantees. Finally, we present simulation data and show that they are consistent with the analytical results.

References

1. Kurose, J., Ross, K.: Computer networking: a top-down approach, 4th edn. Addison Wesley, Reading (2007)
2. McKeown, N.: A fast switched backplane for a gigabit switched router. *Business Communications Review* 27(12) (1997)
3. Katevenis, M., Passas, G.: Variable-size multipacket segments in buffered crossbar (CICQ) architectures. In: *IEEE ICC 2005*, Seoul, Korea (May 2005)
4. Kornaros, G.: BCB: a buffered crossBar switch fabric utilizing shared memory. In: *9th EUROMICRO Conference on Digital System Design*, Croatia, August 2006, pp. 180–188 (2006)
5. Mhamdi, L., Kachris, C., Vassiliadis, S.: A reconfigurable hardware based embedded scheduler for buffered crossbar switches. In: *14th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, February 2006, pp. 143–149 (2006)
6. Papaefstathiou, I., Kornaros, G., Chrysos, N.: Using Buffered crossbars for chip interconnection. In: *17th Great Lakes Symposium on VLSI*, Stresa-Lago Maggiore, Italy, March 2007, pp. 90–95 (2007)
7. Yoshigoe, K., Christensen, K., Jacob, A.: The RR/RR CICQ switch: hardware design for 10-Gbps link speed. In: *22nd IEEE International Performance, Computing, and Communications Conference*, Phoenix, AZ, April 2003, pp. 481–485 (2003)
8. He, S., et al.: On Guaranteed Smooth Switching for Buffered Crossbar Switches. *IEEE/ACM Transactions on Networking* 16(3), 718–731 (2008)
9. Magill, B., Rohrs, C., Stevenson, R.: Output-queued switch emulation by fabrics with limited memory. *IEEE Journal on Selected Areas in Communications* 21(4), 606–615 (2003)

10. Mhamdi, L., Hamdi, M.: Output queued switch emulation by a one-cell-internally buffered crossbar switch. In: IEEE GLOBECOM 2003, San Francisco, CA (December 2003)
11. Stephens, D., Zhang, H.: Implementing distributed packet fair queueing in a scalable switch architecture. In: IEEE INFOCOM 1998, San Francisco, CA (March 1998)
12. Chuang, S., Iyer, S., McKeown, N.: Practical algorithms for performance guarantees in buffered crossbars. In: IEEE INFOCOM 2005, Miami, FL (March 2005)
13. Pan, D., Yang, Y.: Providing flow based performance guarantees for buffered crossbar switches. In: IEEE IPDPS 2008, Miami, FL (April 2008)
14. Turner, J.: Strong performance guarantees for asynchronous crossbar schedulers. IEEE/ACM Transactions on Networking (to appear, 2009)
15. Rojas-Cessa, R., Oki, E., Jing, Z., Chao, H.: CIXB-1: Combined input-once-cell-crosspoint buffered switch. In: IEEE HPSR 2001, Dallas, TX (July 2001)
16. Rojas-Cessa, R., Oki, E., Chao, H.: CIXOB-k: Combined input-crosspoint-output buffered packet switch. In: IEEE Globecom 2001, San Antonio, TX (November 2001)
17. Mhamdi, L., Hamdi, M.: MCBF: a high-performance scheduling algorithm for buffered crossbar switches. IEEE Communications Letters 7(9), 451–453 (2003)
18. Zhang, X., Bhuyan, L.: An efficient scheduling algorithm for combined-input-crosspoint-queued (CICQ) switches. In: IEEE Globecom 2004, Dallas, TX (November 2004)
19. Katevenis, M., Passas, G., Simos, D., Papaefstathiou, I., Chrysos, N.: Variable packet size buffered crossbar (CICQ) switches. In: Proc. IEEE ICC 2004, Paris, France (June 2004)
20. Pan, D., Yang, Y.: Localized independent packet scheduling for buffered crossbar switches. IEEE Transactions on Computers 58(2), 260–274 (2009)
21. Parekh, A., Gallager, R.: A generalized processor sharing approach to flow control in integrated services networks: the single node case. IEEE/ACM Trans. Networking 1(3), 344–357 (1993)
22. Pan, D., Yang, Y.: Max-min fair bandwidth allocation algorithms for packet switches. In: IEEE IPDPS 2007, Long Beach, CA (March 2007)
23. Hosaagrahara, M., Sethu, H.: Max-min fairness in input-queued switches. IEEE Transactions on Parallel and Distributed Systems 19(4), 462–475 (2008)
24. McKeown, N., Mekkittikul, A., Anantharam, V., Walrand, J.: Achieving 100% throughput in an input queued switch. IEEE Trans. Commun. 47(8), 1260–1267 (1999)
25. Bennett, J., Zhang, H.: WF2Q: worst-case fair weighted fair queueing. In: IEEE INFOCOM 1996, San Francisco, CA (March 1996)
26. Leonardi, E., Mellia, M., Neri, F., Marsan, M.: On the stability of input-queued switches with speed-up. IEEE/ACM Trans. Networking 9(1), 104–118 (2001)
27. Farleigh, C., et al.: Packet-level traffic measurements from the Sprint IP backbone. IEEE Network 17(6), 6–16 (2003)

QShine 2009

**Invited Session I – Resource
Management in Wireless Networks**

Joint Optimization of System Lifetime and Network Performance for Real-Time Wireless Sensor Networks

Lei Rao^{1,2}, Xue Liu², Jian-Jia Chen³, and Wenyu Liu¹

¹ Department of EI, Huazhong University of Science and Technology, Wuhan, China
liuwy@mail.hust.edu.cn

² School of Computer Science, McGill University, Montreal, Canada
{leirao,xueliu}@cs.mcgill.ca

³ Computer Engineering and Networks Laboratory, ETH, Switzerland
jchen@tik.ee.ethz.ch

Abstract. Maximizing the aggregate network utility and minimizing the network energy consumption are important but conflict goals in wireless sensor networks. Challenges arise due to the application-specific computing and communication resources constraints and end-to-end real-time constraints. This paper studies the tradeoff between energy consumption and network performance in Real-Time Wireless Sensor Networks (RTWSN) by investigating the interaction between the network performance optimization and network lifetime maximization problems. We address the tradeoff between these two conflict goals as a joint non-linear optimization problem. Based on the solution of the optimization problem, we design an online distributed algorithm to achieve judicious tradeoff based on application-specific focus, while at the same time meeting real-time and resource constraints. Extensive simulation studies illustrate the efficiency and efficacy of the proposed algorithm.

Keywords: Real-Time, Wireless Sensor Networks, System Lifetime, Network Performance.

1 Introduction

Over the last few years, the design of wireless sensor networks has gained increasing importance due to their potential for many military and civil applications, such as fire monitoring, border surveillance, medical care, highway traffic coordination, etc. Many of those applications have real-time requirements. To satisfy the real-time requirements of many of these applications, Real-Time Wireless Sensor Networks (RTWSNs) have been developed in the literature [2] [9] [16][17][18][25].

To design an RTWSN, one has to consider the following issues: network performance, energy consumption, and real-time requirements. Network performance refers to the level of quality of service (QoS) provided by an RTWSN. For energy consumption, since sensor nodes are typically driven by batteries, they have to comply with limited battery budgets. Very often, battery recharging manually or replacement is impossible due to the deployment of sensor nodes in inaccessible or hostile environment. Such energy constraints bring about the notion of network

“lifetime”. The network is considered to be alive while all the nodes still have some energy; the lifetime is the time from the initialization of the network to the time until the first node runs out of energy. For many applications mentioned above, the sensor data are only valid for a limited duration; hence need to be delivered within certain real-time constraints (e.g., deadlines). In this paper, we study problems in RTWSNs considering all of the above three metrics together.

For a specific application in RTWSNs, phenomena of interest will change with time. Instead of using worst-case sampling rates that will work for all the phenomena, dynamically finding the set of globally optimal sampling rates will provide better QoS. For most applications, higher sampling rates offer better QoS. We will exploit network aggregate utility over sampling rates and route selection to refer to the network performance.

In an RTWSN, we can increase the network lifetime or the network utility by using transmission schemes as dynamic routing selection and dynamically finding the set of optimal sampling rates under real-time constraints. We note that there is an inherent tradeoff between the network lifetime and the network utility. Maximizing network utility may require some nodes to lie on routes of many source destination pairs with higher sampling rates, which may cause them to run out of energy quickly. We can increase the network lifetime routing selection avoiding the creation of hot spots where some nodes die out quickly and cause the network to fail as well as decreasing the sampling rates at the expense of the network utility.

In this paper, we systematically study the joint optimization of system lifetime and network utility under schedulability and real-time constraints in an RTWSN. We first model and formulate the lifetime maximization and network utility maximization as two optimization problems separately. By introducing a designed parameter, we formulate the tradeoff between lifetime and network utility as a joint optimization problem. Due to the distributed nature of wireless sensor networks, we design a distributed algorithm for the joint optimization problem. We exploit the NUM framework as discussed in Shu et al.’s work [2]. In this paper, we propose a distributed algorithm for our optimization problem. We will illustrate the tradeoff between system lifetime and network utility by assigning different values to the designed parameter. The designed parameter is selected by domain experts. We will also show that the distributed algorithm is effective in both routes selection and sampling rates assignment.

In summary, the main contribution of this paper is twofold:

- 1) To the best of our knowledge, our study is the first to investigate the joint optimization of system lifetime and network utility under real-time constraints in a wireless sensor network. With different designed parameters for different applications, different routes selection and sampling rates are obtained to meet requirements from those applications.
- 2) We design a distributed algorithm that corresponds to the mathematical solution of our optimization problem, and show the efficiency of the algorithm through simulations.

The rest of this paper is organized as follows: Section 2 briefly introduces the related work; Section 3 formulates and models the joint optimization between system lifetime and network utility; Section 4 solves the optimization problem using the primal-dual

method and dual decomposition technique; Section 5 presents the distributed algorithm that matches the solution in Section 4; Section 6 evaluates the distributed algorithm and tradeoff problem; and Section 7 concludes the paper.

2 Related Work

Due to the wide deployment of wireless sensor networks, many researchers are now developing technologies such as localization, topology control, and power management for different kinds of applications. A comprehensive survey is referred to [10].

Energy efficiency routing algorithms for wireless sensor networks have attracted considerable attentions. Many of the works on this topic focus on minimizing the total energy consumption of the network [11][12][13][24][26][27][28]. Such optimization goal can lead to draining some nodes' energy very quickly. To remedy, [14] proposes the heuristics of selecting routes in a distributed manner to maximize the network lifetime. Distributed iterative algorithms for computing the maximum lifetime flow are described in [15], where each iteration involves a bisection search on the network lifetime and the solution of a max-flow problem to check the feasibility of the network lifetime.

Sha et al. [16] first study the problem of finding the optimal task execution rates subject to the schedulability constraints for digital controllers. To the best of our knowledge, works by Caccamo et al. [17] first deal with multi-hop RTWSN. Adopting their cellular base station backbone layout, Liu et al. design the Real-time Independent Channels (RICH) architecture [9]; model the real-time sampling rate assignment problem as a constrained optimization problem; and propose a distributed algorithm using Internet pricing schemes [18]. In contrast to Liu et al.'s work, Shu et al. [2] consider dynamic routing: in their model, each sensor source has one or more paths leading to its destination, and one path at a time is selected for data transmission. They show their proposed data transmission scheme outperforms the scheme in [9].

Although both network lifetime maximization and utility maximization have been extensively studied in recent years, few works consider them together and study the tradeoff between them in wireless sensor networks [19][20]. To the best of our knowledge, this paper is the first to study such tradeoff.

The problem of resource allocation for congestion control in computer networks is first studied by Kelly [4][5] and Low et al.[18], which lay the foundation of Network Utility Maximization (NUM). Recent researches model the overall communication network as a generalized NUM problem; each layer corresponds to a decomposed sub-problem; and the interfaces among layers are quantified as functions of optimization variables coordinating the sub-problems. The approaches of Chen [22], and Lin [23] show (from different aspects) how a joint optimization problem can be decoupled and separated into different network layers. Based on the primal-dual method and dual decomposition technique from the NUM framework, we solve the optimization problem in this paper, and find an effective distributed algorithm.

In terms of performance metric, we adopt the utility loss index to capture the performance loss. For more details, refer to [9]. RICH architecture is shown in Fig. 1

proposed by Liu et al. employs the mixed FDMA-CDMA scheme. Our work exploits the RICH architecture [9] to support real-time flows on wireless sensor networks. Schedulability analysis is similar to [2]. Due to space limit, interested readers are referred to [2] for details.

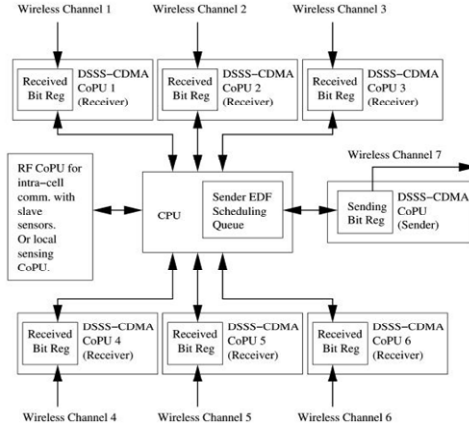


Fig. 1. The Internal Architecture of a RICH Base Station

3 Formulation and Modeling

In an RTWSN, a number of sensor nodes are deployed according to the RICH architecture [9]. We consider about non-rechargeable and irreplaceable sensor batteries. There are multiple source and destination pairs, and there is a set of paths for each source and destination pair, which can be chosen offline based on given routing algorithms for wireless sensor networks such as SPIN [4], GPSR [5], GEAR [6], SPEED [7] or RPAR [8]. The performance of the RTWSN is characterized by two metrics: the network lifetime and the network utility. The network utility is roughly proportional to the allocated sampling rate at each source node.

3.1 Supporting Multi-hop RTWSN

We assume that our wireless base stations (the so-called RICH base stations) have the internal architecture illustrated by Fig. 1. Each RICH base station has seven DSSS-CDMA modulation/demodulation co-processors. Each co-processor operates with a distinct *Direct Sequence Spread Spectrum* CDMA (DSSS-CDMA) *Pseudo Noise* sequence at a distinct FDMA RF band. Six of the DSSS-CDMA co-processors are receivers, and the other one is the only transmitter of the base station.

The broadcast of a RICH base station is overheard by its six neighbor base stations. The wireless medium to the six neighbors are usually. According to DSSS-CDMA theory, given RF band, the upper bound of data bit bandwidth is determined, which we call affordable bandwidth. Suppose for a base station X , due to the irregularity of wireless medium, the affordable bandwidths to its six neighboring RICH base stations

are B_1, B_2, \dots, B_6 . The transmission data bandwidth of X is $B = \min\{B_1, B_2, \dots, B_6\}$. Therefore the broadcast of X is reliably received by all its six neighbors, i.e. B models factors such as the impact of radio irregularity on the wireless medium.

Consider an RTWSN with N nodes, among which there are S sources. Each source s has K^s available paths or routes from the source to its destination. Only one route is selected for transmitting the data for source s . Let f_s be the sampling rate of source s , and each node on the route of source s also forwards data at such rate.

Physical limitations of a sensor imply an upper bound on its sampling rate. On the other hand, an RTWSN application may require a minimum sampling rate to maintain a minimum performance. Hence we have:

$$f^{\min} \leq f \leq f^{\max}, \tag{1}$$

where f is the vector of sampling rates of all sources.

To meet the real-time requirement of an RTWSN, we explore non-preemptive EDF scheduling constraint as our schedulability constraint. For details, refer to [9]. We have:

$$Af \leq b, \tag{2}$$

where matrix A corresponds to the routing topology and is obtained by node-wise analysis, column matrix b reflects the bandwidth of each node in the network.

3.2 Lifetime Maximization

To maximize RTWSN lifetime by joint rate assignment and dynamic route selection, we can formulate the problem as a nonlinear optimization problem with linear constraints.

For each node i , let d_{is} denote the ternary value which is decided by the network topology:

$$d_{is} = \begin{cases} 2, & \text{if } i \text{ both receives and transmits packets from } s; \\ 1, & \text{if } i \text{ only receives or transmits packets from } s; \\ 0, & \text{otherwise.} \end{cases}$$

Here we assume without loss of generality that the wireless transmission energy cost is the same as reception energy cost. Our analysis can be easily extended to the cases where these two costs are not equal.

Each node i is assumed to have initial battery energy E_i . The energy spent by node i to transmit or receive a unit of information is e_i . Then the lifetime of each node i is given by

$$T_i = \frac{E_i}{\sum_{s \in S} d_{is} e_i f_s}. \tag{3}$$

We define the network lifetime T_{net} to be the time until the first node runs out of energy, i.e.:

$$T_{net} = \min_{i \in N} T_i. \tag{4}$$

Our aim is to find an algorithm that maximized the network lifetime. Hence we introduce the network lifetime T_{net} as our objective function.

To find an algorithm that maximizes the network lifetime both under the constraints in (1) and (2), the following optimization problem should be solved:

$$\begin{aligned}
 & \max_{f,R} T_{net} \\
 & \text{subject to} \\
 & f \leq f^{\max}, \\
 & f \geq f^{\min}, \\
 & Af \leq b.
 \end{aligned} \tag{5}$$

Here R refers to all available routes.

The problem in (5) can be re-written as:

$$\begin{aligned}
 & \max_{f,R} T \\
 & \text{subject to} \\
 & f \leq f^{\max}, \\
 & f \geq f^{\min}, \\
 & Af \leq b, \\
 & T \sum_{s \in S} d_{is} e_i f_s \leq E_i, \forall i \in N,
 \end{aligned} \tag{6}$$

where the last set of constraints models the energy consumption at each node. Let $q = \frac{1}{T}$, and $P_{is} = d_{is} e_i$, we obtain an equivalent linear programming formulation:

$$\begin{aligned}
 & \min_{f,R} q \\
 & \text{subject to} \\
 & f \leq f^{\max}, \\
 & f \geq f^{\min}, \\
 & Af \leq b, \\
 & \sum_{s \in S} P_{is} f_s \leq q E_i, \forall i \in N.
 \end{aligned} \tag{7}$$

3.3 Network Utility Maximization

The network utility maximization can be achieved by minimizing the network Utility Loss Index (ULI), which is used to capture the performance loss to the ideal case. According to [9], the ULI has the following general form:

$$U_s(f_s) = \omega_s \alpha_s e^{-\beta_s f_s},$$

where non-negative values ω_s , α_s and β_s are application-specific parameters, which can be determined through curve fitting using measured data.

By employing the ULI as our objective function, the optimization problem for network utility maximization based on joint dynamic routing selection and rate assignment under the constraints in (1) and (2) can be stated as follows:

$$\begin{aligned}
 & \min_{f,R} \sum_s U_s(f_s) \\
 & \text{subject to} \\
 & f \leq f^{\max}, \\
 & f \geq f^{\min}, \\
 & Af \leq b.
 \end{aligned} \tag{8}$$

3.4 Joint Network Lifetime Maximization and Utility Maximization

As stated above, we have two important but conflicting objectives when optimizing the network performance, i.e., achieving network lifetime maximization (5) and maximizing network utility among sensor nodes (8), both of which can be formulated as constrained minimization problem. Hence the tradeoff between them can be formulated as a joint programming problem by introducing the weighting method. In conclusion, we have the joint optimization problem for network lifetime maximization and utility maximization by introducing a designed parameter ω :

$$\begin{aligned}
 & \min_{f,R} \omega q + (1 - \omega) \sum_s U_s(f_s) \\
 & \text{subject to} \\
 & f \leq f^{\max}, \\
 & f \geq f^{\min}, \\
 & Af \leq b, \\
 & \sum_{s \in S} P_s f_s \leq q E_i, \forall i \in N.
 \end{aligned} \tag{9}$$

The designed parameter is based on the requirements from real applications. ω and $1 - \omega$ demonstrate the weights of system lifetime and network utility separately. ω is selected by domain experts.

3.5 An Illustrating Example

To help the readers understand the algorithm better, we give a practical network topology as an example.

Consider the sensor network shown in Fig. 2, where nodes 1 and 3 are sources (numbered s_1, s_2 respectively) that send data to their corresponding destinations 9 and 10 (numbered d_1, d_2 respectively). Using a given routing algorithm, we can obtain the following candidate routes between the sources and the corresponding destinations:

$$\begin{aligned}
 (s_1 \rightarrow d_1) &= \begin{cases} 1 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 9 \\ 1 \rightarrow 2 \rightarrow 4 \rightarrow 7 \rightarrow 9 \end{cases}, \\
 (s_2 \rightarrow d_2) &= \begin{cases} 3 \rightarrow 4 \rightarrow 7 \rightarrow 10 \\ 3 \rightarrow 5 \rightarrow 7 \rightarrow 10. \\ 3 \rightarrow 5 \rightarrow 8 \rightarrow 10 \end{cases}
 \end{aligned}$$

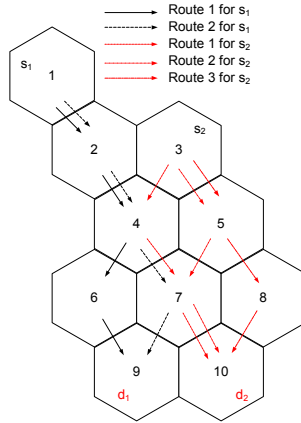


Fig. 2. The Network Topology

If every source s selects the first path as its route, then we can write the corresponding routing set as follows:

$$R = \begin{cases} 1 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 9 \\ 3 \rightarrow 4 \rightarrow 7 \rightarrow 10 \end{cases} .$$

We can define both an $N \times S$ traffic matrix T to specify the relationship between routers and sources, and an $N \times S$ energy matrix P to specify the relationship between nodes energy consumptions and sources. The corresponding traffic matrix and energy matrix for the above R are:

$$T = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 1 \times e_{1,1} & 0 \\ 2 \times e_{2,1} & 0 \\ 0 & 1 \times e_{3,2} \\ 2 \times e_{4,1} & 2 \times e_{4,2} \\ 0 & 0 \\ 2 \times e_{6,1} & 0 \\ 0 & 2 \times e_{7,2} \\ 0 & 0 \\ 1 \times e_{9,1} & 0 \\ 0 & 1 \times e_{10,2} \end{pmatrix} .$$

The parameters of sources are defined in Table 1 and the parameters of nodes are defined in Table 2. The unit for packet size is Kb, the unit for sampling rate is Hz, and the unit for bandwidth is Mbps.

Table 1. Parameters of the data sources of the example

s	p_s	f_s^{\max}	f_s^{\min}	α_s	β_s	ω_s
1	10	11	30	0.33	0.3	4
2	15	2.5	25	0.22	0.2	3

Table 2. Parameters of the nodes of the example

n	1	2	3	4	5
B_n	0.25	0.6	0.4	0.7	0.3
n	6	7	8	9	10
B_n	0.3	0.8	0.15	0.3	0.4

Table 3. Parameters of the nodes energy consumptions of the example

n	1	2	3	4	5
e_n	0.001	0.001	0.001	0.005	0.001
n	6	7	8	9	10
e_n	0.001	0.001	0.005	0.001	0.001

The parameters of the nodes energy consumptions are defined in Table 3. We assume that each node in the network is charged with a 1000 A-hr battery. For sending or receiving a packet with fixed maximum packet length l once, the energy consumption for each sensor node is e_n A-hr. The fixed packet length is assumed to be 1 kb.

4 Joint System Lifetime and Network Utility Optimization

Due to the distributed nature of wireless sensor networks, the above optimization problem should be solved in a distributed manner, and can be interpreted as minimizing the maximum ratio of power consumption to energy supply as well as minimizing the utility loss index at a node. The recent research of Network Utility Maximization (NUM) formulates network system design problem as maximization of the aggregate utility of all the nodes subject to physical or economic constraints, and takes advantage of many advances in nonlinear optimization theory and distributed algorithm. In this section, we solve the optimization problem based on primal-dual method and dual decomposition techniques. We also modify the objective function of the joint optimization problem to meet the distributed computation requirement.

We first form the Lagrangian of the optimization problem as follows:

$$\begin{aligned}
 L(q, f, \lambda, \mu) &= \omega q + (1 - \omega) \sum_{s \in S} U_s(f_s) \\
 &\quad + \sum_{i \in N} \lambda_i \{ \sum_s P_{is} f_s - q E_i \} \\
 &\quad + \sum_{i \in N} \mu_i \{ A_i f - b_i \} \\
 &= \omega q + (1 - \omega) \sum_{s \in S} U_s(f_s) \\
 &\quad + \sum_{i \in N} \lambda_i \{ \sum_s P_{is} f_s - q E_i \} \\
 &\quad + \sum_{i \in N} \{ f_s \sum_s A_{si} \mu_i - \mu_i b_i \},
 \end{aligned} \tag{10}$$

where Lagrangian multipliers λ_i are introduced for the energy constraint while μ_i are introduced for the scheduling condition. Then the dual problem is

$$D(\lambda, \mu) = \inf_{f^{\min} \leq f \leq f^{\max}, q \geq 0} L(q, f, \lambda, \mu). \tag{11}$$

Since the objective function is not strictly convex in the primal variables, the dual function is non-differential. In this case, we change the primal objective function to $\omega \sum_{i \in N} q_i^2 + (1 - \omega) \sum_{s \in S} U_s(f_s)$. For the detailed discussions, refer [3]. Define the two nodes sharing the same link in the wireless sensor network as neighbors to each other. Hence for a node i , we can define a set of nodes which are neighbors of i as its neighbor set N_i . Then we have the optimization problem as:

$$\begin{aligned}
 &\min_{f, R} \omega \sum_{i \in N} q_i^2 + (1 - \omega) \sum_{s \in S} U_s(f_s) \\
 &\text{subject to} \\
 &\quad f \leq f^{\max}, \\
 &\quad f \geq f^{\min}, \\
 &\quad Af \leq b, \\
 &\quad \sum_{s \in S} P_{is} f_s \leq q_i E_i, \forall i \in N, \\
 &\quad q_i = q_j, \forall i \in N, j \in N_i.
 \end{aligned} \tag{12}$$

Hence the dual function is differentiable. Re-write the Lagrangian for the regularized objective function, we have:

$$\begin{aligned}
 L(q, f, \lambda, \mu, \gamma) &= - \sum_{i \in N} \mu_i b_i + (\omega \sum_{i \in N} q_i^2 - \sum_{i \in N} q_i \lambda_i E_i) \\
 &\quad + \sum_s ((1 - \omega) U_s(f_s) + f_s \sum_{i \in N} (\mu_i A_{si} - \lambda_i P_{is})) \\
 &\quad + \sum_{i \in N} \sum_{j \in N_i} \gamma_{ij} (q_i - q_j)^2,
 \end{aligned} \tag{13}$$

from which it is clear that the dual function can be evaluated separately in each of the variables f_s and q_i . The dual decomposition results in each source s solving, for the given λ , μ and γ , at each iteration t :

$$f_s^{(t+1)} = \arg \min_{f_s^{\min} \leq f_s \leq f_s^{\max}} ((1 - \omega)U_s(f_s) + f_s (\sum_{i \in N} \mu_i^{(t)} A_{si} - \lambda_i^{(t)} P_{is})), \quad (14)$$

$$R^s(t+1) = \arg \min_{R^s \in H^s} \sum_{i \in N} (\mu_i(t) A_{si} + \lambda_i(t) P_{is}), \quad (15)$$

and in each node i solving, for the given λ , μ and γ , at each iteration t :

$$q_i^{(t+1)} = \arg \min_{q_i} (\omega q_i^2 - q_i \sum_{i \in N} \lambda_i^{(t)} E_i + q_i^2 \sum_{j \in N_i} (\gamma_{ij} - \gamma_{ji})), \quad (16)$$

The master dual problem is given by

$$\begin{aligned} \min_{\lambda, \mu, \gamma} \quad & \inf_{f^{\min} \leq f \leq f^{\max}, q \geq 0} L(q, f, \lambda, \mu, \gamma) \\ \text{s.t.} \quad & \lambda \geq 0, \quad \mu \geq 0, \quad \gamma \geq 0. \end{aligned} \quad (17)$$

Using sub-gradient method, the Lagrangian multipliers can be updated as following at each iteration t :

$$\lambda_i^{(t+1)} = [\lambda_i^{(t)} + \alpha_i (\sum_s P_{is} f_s^{(t)} - q_i^{(t)} E_i)]^+, \quad \forall i \in N, \quad (18)$$

$$\mu_i^{(t+1)} = [\mu_i^{(t)} + \alpha_k (A_{si} f_s^{(t)} - b_i)]^+, \quad \forall i \in N, \quad (19)$$

$$\gamma_{ij}^{(t+1)} = [\gamma_{ij}^{(t)} + \alpha_i (q_i - q_j)^2]^+, \quad \forall i \in N, j \in N_i, \quad (20)$$

where α_i is a positive step-size and $[\bullet]^+$ is defined as $[x]^+ = \max\{x, 0\}$.

5 Distributed Algorithm

In this section, we present our design of the distributed algorithm for the optimization problem following the solution in Section 4 as follows:

1) Initialization of the distributed algorithm

- Each node n sets all the energy constraints relevant prices and scheduling constraints relevant prices to 1.
- Each node n adds all the neighbor nodes to its neighbor set and sets $q_n = \text{Inf}$.
- Each source s sets the sampling rate f_s to f_s^{\min} .
- Each source s selects the first candidate route as R .

2) Iteration of the distributed algorithm

In each iteration step t , the prices, sampling rates, lifetime and routes are updated.

Prices update

- The latest rate proposal is sent by each source in a rate proposal packet to corresponding destination along the currently selected route.

b. On receiving rate proposals from all relevant sources, each node n computes new prices for the constraints with the price updating equations (18), (19) and (20).

Lifetime update

- a. On receiving rate proposals from all relevant sources, each node n computes new local lifetime parameter q_n with the lifetime updating equation (16).
- b. Each node n broadcasts both its new local lifetime and new price γ_n to all its neighbors and receives the update lifetime and price message from all its neighbors.

Sampling rates update

- a. A sampling rate update packet with value 0 is sent by each destination along the reserved path of the current route to corresponding source.
- b. Each node n adds $\mu_n A_{ns}$ and $\lambda_n P_{ns}$ to the value in the incoming sampling rate update packet, and forwards it along the reversed path.
- c. On receiving all relevant sampling rate update packets, each source s updates its rate proposal according to local optimization with equation (14).

Routing update

- a. A routing update packet with value 0 is sent by each destination along the reserved path of every possible route to the source.
- b. Each node n adds $\mu_n A_{ns}$ and $\lambda_n P_{ns}$ to the value in the incoming routing update packet, and forwards it along the reversed path.
- c. On receiving all relevant routing update packets, each source s updates routing according to local optimization with equation (15).

6 Performance Evaluations

In this section, we demonstrate the efficacy of our solutions with extensive MATLAB simulation results and some further analysis. The simulation is based on the parameters of the example in Section 4.

6.1 Convergence

First, we conduct a simulation experiment using the distributed algorithm proposed in Section 4 with the step size set to 0.2 and the designed parameter ω set to 0.0. Hence the optimization problem becomes an optimization problem for network utility maximization. The optimal network ULI is 0.0268, with very high q_n , which corresponds to very low network lifetime, $f^* = (15.6250, 19.0476)^T$, and $r^* = (2, 1)$ where the s th element in r^* indicates the optimal route for source s . The convergence of the distributed algorithm is shown in Fig. 3. Fig. 3 shows the convergence of sampling rate of each source.

We also conduct a simulation experiment with $\omega=1.0$, and the optimization problem becomes an optimization problem for network lifetime maximization. The optimal network ULI is 0.4490, with $q_n = 0.0270$, $f^* = (11.000, 2.500)^T$ and $r^* = (2, 2)$.

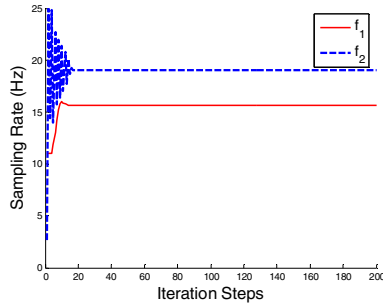


Fig. 3. The Convergence of Sampling rates with $\omega = 0.0$

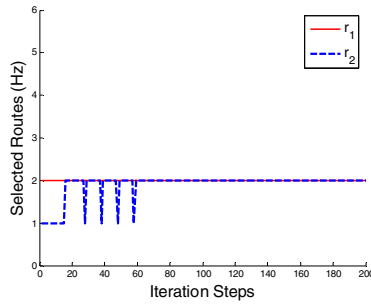


Fig. 4. The Convergence of Routing Selection with $\omega = 0.75$

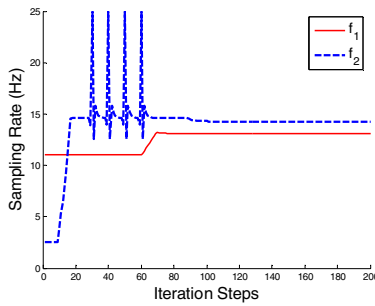
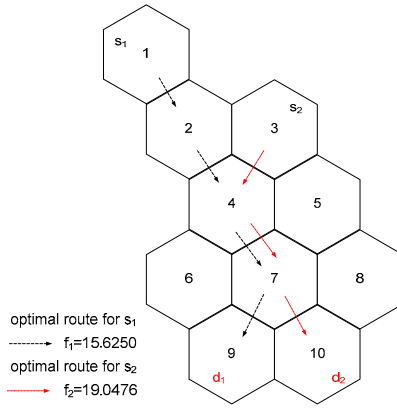
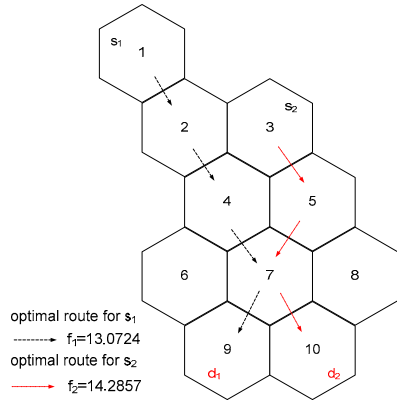


Fig. 5. The Convergence of Sampling Rates with $\omega = 0.75$

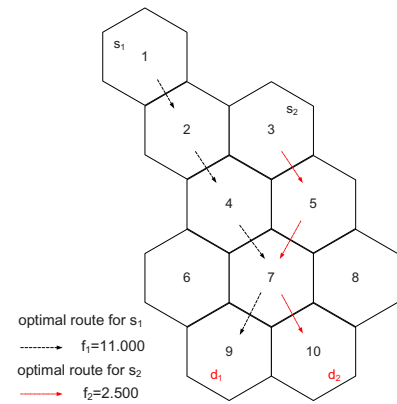
Again, with the designed parameter set to 0.75, the optimal network is 0.0641, with $q_n = 0.1307$, $r^* = (2, 2)$ and $f^* = (13.0724, 14.2857)^T$. The convergence of sampling rate of each source is shown in Fig. 4 while the convergence of routing selection of each source is shown in Fig. 5.



(a)



(b)



(c)

Fig. 6. (a) Sampling Rates and Routes Selection with $\omega=0.0$ (b) Sampling Rates and Routes Selection with $\omega=0.75$ (c) Sampling Rates and Routes Selection with $\omega=1.0$

6.2 Illustrations of Tradeoff between System Lifetime and Network Utility

The optimal sampling rates and routes selection with different designed parameter ω are illustrated in Fig. 6 to demonstrate the necessity of calculating tradeoff between lifetime maximization and utility maximization. In Fig. 6 (a), the designed parameter ω is set to 0.0; hence only network utility maximization is taken into considerations; while in Fig. 6 (b), the designed parameter ω is set to 0.75; hence both network lifetime and utility maximization are taken into considerations. Comparing the results in Fig. 6 (b) against Fig. 6 (a), optimal sampling rates are decreased; hence network utility is decreased and the network ULI is increased from 0.0268 to 0.0641. It is clear that selected routes have changed to avoid the overload of sensor nodes (node 4 in this example) so as to improve the network lifetime. In Fig. 6 (c), the designed parameter ω is set to 1.0 and only network lifetime maximization is considered. Comparing the results in Fig. 6 (b) against Fig. 6 (c), optimal sampling rates are increased to improve the network utility. The above results show that depending on the given application, we can maneuver the designed parameters to adjust both sampling rates and selected routes to achieve the best combined network performance and lifetime.

7 Conclusion and Future Work

In this paper, we investigate the novel problem of the joint optimization of system lifetime and network utility in RTWSNs. We model the joint optimization as a holistic joint optimization problem by deriving the nonlinear objective function and linear constraints. We also introduce a designed parameter to capture the weight balance between network lifetime maximization problem and network utility maximization problem. An effective distributed algorithm is developed based on the primal-dual method and dual decomposition technique. We show the fast convergence and the efficacy of the distributed algorithm, via extensive simulation studies. We demonstrate the necessities of the joint optimization studies in RTWSNs by analyzing and comparing the numerical results with different values of the designed parameter.

As a next step work, we plan to carry out experiments in the real sensor test bed to evaluate our proposed method in this paper.

Acknowledgments. This work was supported in part by NSERC Discovery Grant 341823-07, NSERC Strategic Grant STPGP 364910-08, FQRNT Grant 2010-NC-131844 and Chinese National 863 project (No. 2007AA01Z223).

References

- [1] Caccamo, M., Zhang, L.Y., Sha, L., Buttazzo, G.: An implicit prioritized access protocol for wireless sensor networks. In: Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS), pp. 39–48 (2002)
- [2] Shu, W., Liu, X., Gu, Z., Gopalakrishnan, S.: Optimal sampling rate assignment with dynamic route selection for real-time wireless sensor networks. In: Proceedings of the 29th IEEE Real-Time Systems Symposium (RTSS 2008), Barcelona, Spain (2008)

- [3] Madan, R., Lall, S.: Distributed algorithm for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Wireless Communications* 5(8) (August 2006)
- [4] Heinzelman, W., Kulik, J., Balakrishnan, H.: Adaptive protocols for information dissemination in wireless sensor networks. In: *Proceedings of the 5th International Conference on Mobile Computing and Networking (MobiCom)*, pp. 174–185 (1999)
- [5] Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless sensor networks. In: *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom)*, pp. 243–254 (2000)
- [6] Xu, Y., Heidemann, J.S., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In: *Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom)*, pp. 70–84 (2001)
- [7] He, T., Stankovic, J.A., Lu, C., Abdelzaher, T.: SPEED: a stateless protocol for real-time communication in sensor networks. In: *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 46–55 (2003)
- [8] Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J., Abdelzaher, T.: Real-time power-aware routing in sensor networks. In: *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS)*, pp. 83–92 (2006)
- [9] Liu, X., Wang, Q., He, W., Caccamo, M., Sha, L.: Optimal realtime sampling rate assignment for wireless sensor networks. *ACM Transactions on Sensor Networks* 2(2), 263–295 (2006)
- [10] Akyildiz, I.F., Melodia, T., Chowdhury, K.: *Wireless Multimedia Sensor Networks: A Survey*. *IEEE Wireless Communications Magazine* 14(6), 32–39 (2007)
- [11] Rodoplu, V., Meng, T.H.: Minimum energy mobile wireless networks. *IEEE J. Select. Areas Communi.* 17(8), 1333–1344 (1999)
- [12] Wattenhofer, R., Li, L., Bahl, P., Wan, Y.: Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In: *IEEE INFOCOM* (2001)
- [13] Li, L., Halpern, J.Y., Bahl, P., Wang, Y., Wattenhofer, R.: Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In: *ACM Symposium on Principle of Distributed Computing, PODC* (2001)
- [14] Chang, J.H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: *IEEE INFOCOM*, pp. 22–31 (2000)
- [15] Zussman, G., Segall, A.: Energy efficient routing in ad hoc disaster recovery networks. In: *INFOCOM* (2003)
- [16] Sha, L., Liu, X., Caccamo, M., Buttazzo, G.: Online control optimization using load driven scheduling. In: *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 5, pp. 4877–4882 (2000)
- [17] Buttazzo, G., Caccamo, M., Zhang, L.Y., Sha, L.: An implicit prioritized access protocol for wireless sensor networks. In: *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, pp. 39–48 (2002)
- [18] Low, S.H., Lapsely, D.E.: Optimization flow control. I. Basic algorithm and convergence. *IEEE/ACM Transactions on Networking* 7, 861–874 (1999)
- [19] Nama, H., Chiang, M., Mandayam, N.: Utility-lifetime trade-off in self-regulating wireless sensor networks: A cross-layer design approach. In: *Proc. IEEE ICC* (2006)
- [20] Zhu, J., Chen, S., Bensaou, B., Hung, K.L.: Tradeoff between lifetime and rate allocation in wireless sensor networks: a cross layer approach. In: *IEEE INFOCOM 2007*, pp. 267–275 (2007)
- [21] Kelly, F.: Charging and rate control for elastic traffic: Focus on elastic services over ATM networks. *European transactions on telecommunications* 8, 33–37 (1997)

- [22] Chen, L., Low, S.H., Chiang, M., Doyle, J.C.: Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In: IEEE INFOCOM (2006)
- [23] Lin, X., Shroff, N.B.: The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Transactions on Networking* 14(2), 1804–1814 (2006)
- [24] Chen, J., Kuo, T., Lu, H., Yang, C., Pang, A.: Dual power assignment for network connectivity in wireless sensor networks. In: IEEE Global Telecommunications Conference (GlobeCom)
- [25] Li, M., Liu, Y.: Underground Coal Mine Monitoring with Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* 5(2) (March 2009)
- [26] Moser, C., Chen, J., Thiele, L.: Power Management in Energy Harvesting Embedded Systems with Discrete Service Levels. In: The International Symposium on Low Power Electronics and Design (ISLPED) (2009)
- [27] Moser, C., Chen, J., Thiele, L.: Reward Maximization for Embedded Systems with Renewable Energies. In: The 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA (2008)
- [28] Hsiu, P., Wu, C., Kuo, T.: Maximum-Residual Multicasting and Aggregating in Wireless Ad-Hoc Networks. Accepted and to appear in *ACM/Springer Wireless Networks*

Network-Assisted Radio Resource Management for Cell-Edge Performance Enhancement

Young-June Choi¹, Narayan Prasad², and Sampath Rangarajan²

¹ Ajou University, Suwon, 443-749, Korea

² NEC Laboratories America, Princeton, NJ 08540, USA

Abstract. A number of network-level techniques have been proposed to mitigate inter-cell interference and improve throughput for cell-edge users in wide-area wireless data networks. To facilitate the coordination among base stations (BSs), we propose a new radio-resource management framework where cell-edge users and cell-interior users are separately managed by two different radio-resource managers (RRM). In the proposed framework, we address the issue of how to classify a user as cell-edge user or cell-interior user, and how much radio resource the cell-edge users may occupy. We present a solution where a user switches the RRM so as to maximize overall network throughput subject to the condition that her own throughput does not decrease upon switching. We verify our solution using analysis and simulation experiments, and demonstrate that our solution can guarantee superior cell-edge performance and achieve high network throughput.

1 Introduction

In OFDMA systems, as neighboring cells can reuse the same frequency, intercell interference is an important problem that needs to be solved. Due to intercell interference, *cell-edge users* may suffer from high error rates (and hence a reduced throughput) even when the most robust modulation and coding techniques are used. To enhance the performance of cell-edge users in OFDMA systems, frequency reuse techniques between neighboring cells, have been developed. For flexibility, dynamic fractional frequency reuse (FFR) has been widely examined [1, 2, 3], and it is known that FFR can enhance cell-edge throughput by about 15% [4] but at the expense of a reduced average cell throughput. Another technique to enhance cell-edge user performance is macro-diversity. With macro-diversity, multiple BSs can serve a user, thus making the link condition of cell-edge users more reliable and robust [5].

Both dynamic FFR and macro-diversity require coordinated RRM¹ between neighboring BSs within the network. If dynamic FFR is deployed in the system, designated neighboring BSs of a cell to which a certain cell-edge user is attached, should avoid concurrent transmission over the set of channels assigned to that user. On the other hand, if macro-diversity is used, one or more neighboring BSs should serve a certain cell-edge user at the same time; this means

¹ We use RRM to refer to both Radio Resource Management and Radio Resource Managers.

concurrent transmissions over the same set of channels from multiple BSs to the same user is required. Network-level coordination of radio-resources through *Network-Assisted RRM* is required to handle such requirements.

In this paper, we propose a *two-level RRM* framework. We advocate the coexistence of two RRM entities, an upper-level RRM and a lower-level RRM, within the backhaul architecture that connects the BSs. We separate users attached to a BS into two groups; one group consists of users who are classified as cell-edge users and the other consists of users who are classified as cell-interior users. The RRM functions for cell-edge users are handled by the upper-level RRM, whereas those for cell-interior users are handled by the lower-level RRM.

The classification of cell-edge and cell-interior users are *not* based purely on geographic location as in conventional frequency reuse techniques. We classify users as cell-edge or cell-interior users with the goal of maximizing network throughput subject to certain conditions on the per-user throughput; for example, a user at the edge of a cell may still get classified as a cell-interior user if such classification leads to higher network throughput with no attendant loss in the user throughput or conversely if such classification increases the user throughput² without any noticeable loss in the network throughput. Furthermore, we also show that compared to a switching scheme which only aims to maximize the network throughput, our classification scheme results in a better cell-edge performance without a loss in network throughput. Within the proposed framework, we address three main problems: 1) initial user classification, 2) strategy to switch users from one class to another subsequently, and 3) radio-resource reservation in neighboring cells for users who are classified as cell-edge users.

2 Initial User Classification and Metrics

In this section, we introduce metrics for classifying a user as a cell-edge user to be assigned to an upper RRC, and also describe the initial classification of a new user.

2.1 Computation of User Capacities

We would like to determine the capacity a user can achieve when it belongs to the lower RRC (cell-interior user group) or an upper RRC (cell-edge user group). For simplicity, we suppose that a user is able to measure her average signal-to-interference-plus-noise ratio (SINR) as well as the average signal strengths from one dominant neighboring BS and her serving BS, respectively. In the following, we only deal with the case where a cell-edge user is served by at most two BSs, since it is easy to extend the analysis to the case where three or more BSs can serve the user.

Capacity of cell-interior users. Consider a specific lower RRC user in cell 1 and assume that it is served only by BS 1 without any cooperation by neighboring

² This can happen due to the multi-user diversity gain which is obtained when channel dependent scheduling is employed to serve cell interior users.

BSs. Let C_1 denote the resulting downlink capacity per unit resource and let S_1 denote the average received signal strength from BS 1 at the user of interest. Further, let the dominant interfering neighboring BS be indexed by 2 with I_2 denoting the average received signal strength from BS 2 at the user of interest. Next, let I_o be the average interference to the user (which is in cell 1) generated by neighboring BSs other than BS 2 and let N be thermal noise variance. Then, the average received SINR of the user is given by $S_1/(I_2 + I_o + N)$. C_1 is a function of this average SINR. In the case where channel independent scheduling is employed and the users are allocated rates based on their average SINRs, C_1 can be computed as

$$C_1 = \log \left(1 + \frac{S_1}{I_2 + I_o + N} \right). \quad (1)$$

Letting $\gamma_1 = S_1/(I_o + N)$ and $\gamma_2 = I_2/(I_o + N)$, we can rewrite C_1 as

$$C_1 = \log \left(1 + \frac{S_1}{\gamma_2(I_o + N) + I_o + N} \right) = \log \left(1 + \frac{\gamma_1}{\gamma_2 + 1} \right). \quad (2)$$

Note that the average SINR here is a function of the distance dependent path-loss and possibly large scale shadow fading but not of the small scale fading which changes on a much finer time scale and is assumed to be averaged out.

In case opportunistic channel dependent scheduling is employed by the base station, this capacity will increase owing to multiuser diversity. Under some assumptions on the fading process, closed-form approximations for this capacity can be derived using results in [6,7]. Thus, when a cell-edge user tries to switch to the lower RRC, the system can approximate the expected average capacity by such expressions or more generally by using a real-time statistic that is computed from a look up table obtained by measuring user throughputs over some duration in the network.³ Hereafter, C_1 will represent the average capacity of an interior user computed using one of these methods.

Capacity of cell-edge users. Now consider two cases where the cell-edge users are supported by, a) fractional frequency reuse, and b) macro-diversity. In each case no opportunistic scheduling is employed and the users are assigned rates based on their average SINRs.

a) Dynamic FFR – In this case, to mitigate the interference from a dominant neighboring cell at a particular cell-edge user, the two BSs (serving BS as well as the dominant interfering BS) are coordinated such that the dominant neighboring BS will not use a certain quantity of resources that is allocated to the cell-edge user. As interference from the neighboring BS is eliminated, the user can achieve a better capacity. In particular, in the above example, I_2 is removed so the user's capacity achieved by the cooperation of BSs 1 and 2 via FFR, denoted by $C_{1,2}$, is expressed as

$$C_{1,2} = \log \left(1 + \frac{S_1}{I_o + N} \right) = \log(1 + \gamma_1). \quad (3)$$

³ We note that such a look up table is indeed required to implement the proportional fair scheduler.

b) Macro-diversity – We consider Alamouti’s space-time coding [8] for supporting downlink macro-diversity. That is, the serving BS and the dominant neighboring BS transmit two signals y_1 and y_2 at the same time over the same frequency band, followed by $-y_2^*$ and y_1^* . The transmissions from the two BSs can be coherently combined using a simple receiver [8]. Then, if the user is served by an upper RRC for macro-diversity, her capacity will be given by⁴

$$C_{1,2} = \log \left(1 + \frac{S_1 + I_2}{I_o + N} \right) = \log(1 + \gamma_1 + \gamma_2). \tag{5}$$

Obviously, $C_{1,2}$ in the two cases is higher than C_1 given in (1), but some amount of resource from BS 2 needs to be additionally provisioned for this user.

2.2 Computation of Throughput

The throughput of a cell-interior user i in cell x is denoted by $T_x(i)$ and it can be expanded as $T_x(i) = \alpha(i)C_x(i)$, where $C_x(i)$ denotes the average capacity of the interior user i in cell x . $\alpha(i)$ denotes the average ratio of resource allocated to user i (e.g., the average ratio of slots or quantity of resource in the frequency and time domains). Similarly, the throughput of a cell-edge user i managed by the cooperation of BS x and BS y is denoted by $T_{x,y}(i)$ and it can be expanded as $T_{x,y}(i) = \alpha(i)C_{x,y}(i)$, where the average capacity $C_{x,y}(i)$ can be computed as in (3) or (5) depending on whether FFR or macro-diversity is employed. Note that each cell expends a fraction of its available resources to serve the cell-edge users.

The assignment of α ’s relies on a scheduling policy employed at the BSs. We do note that while user k is managed by the lower RRC, $\alpha(k)$ may be adjusted by the scheduling policy used or by the reclassification of other users. On the other hand, the ratio α for a cell-edge user is determined when the user is admitted into the network as a cell-edge user or when the user is switched from the lower RRC to the upper RRC. This computation will be illustrated in the sequel. However, we assume $\alpha(k)$ to be a constant value while user k is being managed by the upper RRC. This simplifying assumption is made because resource rearrangement for such users entails complex calculation involving all the combinations of pairs of neighboring BSs. To summarize, $\alpha(k)$ changes in the following cases.

- $\alpha(k)$ can decrease, if user k is managed by the lower RRC and a new user requiring the cell resource arrives.

⁴ It is possible to obtain orthogonal space-time codes for three transmit antennas, which in our case correspond to the antennas at the three neighboring BSs. From [9], it can be inferred that the resulting capacity is given by

$$C_{1,2,3}(k) = \frac{3}{4} \cdot \log \left(1 + \frac{S_1 + I_2 + I_3}{I'_o + N} \right), \tag{4}$$

where I_2 and I_3 are the received signal strengths from two neighboring BSs, and I'_o is the interference from neighboring BSs other than those two BSs.

- $\alpha(k)$ can increase, if user k is managed by the lower RRC and some resource is freed due to the departure of an existing user who occupied the cell resource.
- $\alpha(k)$ can increase or decrease, if user k switches the serving RRC from a lower RRC to an upper RRC, or vice versa.
- Besides, $\alpha(k)$ is forced to change by a hand-off that occurs regardless of the classification of the user.

Let β_x be the ratio of resource in cell x allocated for cell-edge users. β_x can then be expressed as

$$\beta_x = \sum_{y \in V_x} \sum_{i \in U_{x,y}} \alpha(i), \quad (6)$$

where $U_{x,y}$ is the set of cell-edge users which are managed by the cooperation of BSs x and y , and V_x is the set of BSs which cooperate with BS x (i.e., its neighboring BSs). BS x will use the remaining resource $1 - \beta_x$ for its cell-interior users. We further assume that $\beta_x \leq \beta_{\max}$ in order to avoid monopolization of the resources by the upper RRC.

2.3 Initial User Classification

A new user is admitted to the system as a cell-edge or a cell-interior user. We consider a simple scheme that guarantees a minimum throughput $T_{\min}(i)$ given by user i 's QoS requirement:

$$T_x(i) \geq T_{\min}(i); T_{x,y}(j) \geq T_{\min}(j) \quad \forall x, y, i, j. \quad (7)$$

The capacity of a new user n , $C_x(n)$, upon admission as a cell-interior user in cell x is first estimated. Similarly the capacity $C_{x,y}(n)$ of the user upon admission as a cell-edge user served by BSs x and y is also computed using (3) or (5). Then, the user can be admitted as a cell-interior user by BS x only if its minimum throughput requirement can be met, i.e., only if

$$\sum_{i \in L_x} \frac{T_{\min}(i)}{C_x(i)} + \frac{T_{\min}(n)}{C_x(n)} \leq 1 - \beta_x - \delta, \quad (8)$$

where L_x is the set of cell-interior users in cell x and δ is a margin for absorbing the change of some users' average capacities or accepting hand-off users; for our discussion, δ is considered to be a design parameter. If user n is admissible in BS x as a cell-interior user, a ratio $\alpha(n)$ is determined according to the scheduling policy adopted by BS x . Once $\alpha(n)$ is determined, the admission controller checks if there exists an $\alpha'(n)$ acceptable by BSs x and y for some $y \in V_x$ (using our upward RRC switch algorithm in Section 3) which can lead to better system and user throughputs. If such an $\alpha'(n)$ exists, the user is admitted as a cell-edge user which is served by BSs x and y ; otherwise it is admitted as a cell-interior user which is served by BS x .

Notice that in (8), we have implicitly assumed that the capacity $C_x(i)$ of an existing interior user i in cell x does not change upon addition of a new

user. However when channel dependent scheduling is employed this capacity may increase due to a larger multi-user diversity gain. Thus (8) is a conservative condition for admitting a new user. In general, for channel dependent scheduling, the increase in $C_x(i)$ with the addition of a new user or the decrease in $C_x(i)$ with the deletion of another interior user, is small when the number of interior users is sufficiently large (10 or more verified in simulations). Henceforth, in the case of channel dependent scheduling, we will assume a sufficiently large population of interior users in each cell and ignore this change in the average capacity of an interior user.

3 Strategy for User Reclassification

We now derive the condition for reclassifying users and switching them from upper RRC to lower RRC or vice versa. Users that do not satisfy these conditions will, by default, not be reclassified. The objective behind reclassifying users is to maximize the sum throughput over all the users in the network covered by an ASN gateway (or a set of BSs deployed for cooperation) subject to a minimum throughput guarantee for each user. In particular, the admission controller allocates each user to either a lower RRC or an upper RRC to meet the following objective:

$$\max \left[\sum_{x \in \mathcal{N}} \sum_{i \in L_x} T_x(i) + \sum_{x,y \in \mathcal{N}} \sum_{j \in U_{x,y}} T_{x,y}(j) \right] \tag{9}$$

$$T_x(i) \geq T_{\min}(i); T_{x,y}(j) \geq T_{\min}(j) \forall x, y, i, j.$$

where \mathcal{N} is the set of BSs within the domain. Further, this reclassification is also subject to the condition that the switching (reclassified) user’s throughput must not decrease.

We are now ready to propose our reclassification strategy in which a user is allowed to switch only if both its own throughput as well as the system throughput do not decrease and at-least one of them strictly increases.

3.1 Upward RRC Switch

We first consider an upward RRC switch algorithm, when user k tries to switch her RRC from a lower RRC to an upper RRC. Assume that the user is being served by cell 1 and the current ratio α for the user is $\alpha(k)$. Suppose user k ’s ratio changes to $\alpha'(k)$ after the RRC switch, when she is supposed to be managed by BSs 1 and 2. User k will accept the RRC change when her throughput becomes higher by changing the RRC, so the first condition for reclassification is

$$\alpha'(k)C_{1,2}(k) - \alpha(k)C_1(k) \geq 0. \tag{10}$$

Since $\alpha'(k)C_{1,2}(k) \geq \alpha(k)C_1(k) \geq T_{\min}(k)$, the condition in (10) will ensure that the minimum throughput requirement will also be satisfied post-switching.

Next, we consider the impact of switching on system throughput which is more involved. In particular there are three factors that must be accounted for:

- *The throughput loss in cell 2:* Notice that user k post-switching will take an additional resource $\alpha'(k)$ from BS 2 which could have been used for other users in that cell if it had not been used for dynamic FFR or macro-diversity. However, it is very hard to precisely estimate this throughput loss since it depends on the cell 2's scheduling rule. Consequently, we use a simple way to quantify this loss as $\alpha'(k) \cdot \overline{C}_2$, where \overline{C}_2 is the average per-user capacity of cell 2's interior users⁵.
- *The throughput change in cell 1:* The throughput of the current serving cell (cell 1) can change due to switching in the following manner. First, if $\alpha'(k) < \alpha(k)$, the residual part $\alpha(k) - \alpha'(k)$ will be distributed among cell 1's interior users and together they will achieve an average throughput gain of $(\alpha(k) - \alpha'(k)) \cdot \overline{C}_1$, where \overline{C}_1 is the average per-user capacity of cell 1's interior users (excluding user k). Otherwise, i.e., if $\alpha'(k) > \alpha(k)$, cell 1's interior users will lose an average throughput of $(\alpha'(k) - \alpha(k)) \cdot \overline{C}_1$. In either case, the net throughput change in cell 1 is expressed by $(\alpha(k) - \alpha'(k)) \cdot \overline{C}_1$.
- *System constraints:* We must ensure that the switching operation does not violate the minimum throughput requirement of any user or the maximum limit on the resource ratio reserved for cell-edge users in any cell. Specifically, if either $\beta_1 + \alpha'(k)$ or $\beta_2 + \alpha'(k)$ is greater than β_{\max} , or the additional resource $\alpha'(k)$ taken from BS 2 or $\alpha'(k) - \alpha(k)$ taken from BS 1 (when $\alpha'(k) > \alpha(k)$) jeopardizes the minimum allocation for users in L_2 or $L_1 - \{k\}$, user k cannot be allowed to use $\alpha'(k)$ by the upper RRC.

Thus, the first two conditions dictate that a post-switching ratio $\alpha'(k)$ chosen to maximize the network-side throughput in eq. (9), should satisfy

$$\alpha'(k)[C_{1,2}(k) - \overline{C}_1 - \overline{C}_2] - \alpha(k)[C_1(k) - \overline{C}_1] \geq 0 \quad (11)$$

On the other hand, the system constraints impose that $\alpha'(k)$ should also be constrained to satisfy:

$$\begin{aligned} \alpha'(k) &\leq \min[\beta_{\max} - \beta_1, \beta_{\max} - \beta_2, \\ &1 - \beta_1 - \sum_{i \in L_1 - \{k\}} \frac{T_{\min}(i)}{C_1(i)}, 1 - \beta_2 - \sum_{i \in L_2} \frac{T_{\min}(i)}{C_2(i)}]. \end{aligned} \quad (12)$$

Thus, the optimal ratio $\alpha'(k)$ can be determined by solving the following optimization problem:

$$\begin{aligned} \max \quad & \alpha'(k)[C_{1,2}(k) - \overline{C}_1 - \overline{C}_2] - \alpha(k)[C_1(k) - \overline{C}_1] \\ \text{subject to} \quad & \text{(10), (11) and (12)}. \end{aligned} \quad (13)$$

The solution for the above objective is given by the following proposition which is proved in Appendix A

⁵ Note that with our assumption of infinitely backlogged traffic, cell-interior users of any BS will always fully utilize the available resources.

Proposition 1. *The condition of changing a user k 's RRC from a lower RRC to an upper RRC with the cooperation of BSs 1 and 2 is summarized as follows.*

i) If $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 < 0$ and $C_1(k) - \bar{C}_1 < 0$, then switching is allowed only if

$$\bar{C}_1 \cdot C_{1,2}(k) - (\bar{C}_1 + \bar{C}_2) \cdot C_1(k) \geq 0 \tag{14}$$

and if the post-switching ratio $\alpha(k)C_1(k)/C_{1,2}(k)$ meets the condition (12). The optimal $\alpha'(k)$, when these two conditions are met, is given by

$$\alpha'(k) = \alpha(k)C_1(k)/C_{1,2}(k). \tag{15}$$

ii) If $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 = 0$ and $C_1(k) - \bar{C}_1 < 0$, $\alpha'(k)$ can be chosen arbitrarily subject to (10) and (12).

iii) If $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 > 0$ and $C_1(k) - \bar{C}_1 \leq 0$, $\alpha'(k)$ should be the maximal available value subject to (10) and (12).

The case of $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 > 0$ and $C_1(k) - \bar{C}_1 > 0$ will be separately mentioned at the end of this subsection.

3.2 Downward RRC Switch

Next, we describe a downward RRC switch algorithm, when user k managed by the upper RRC through cooperation between BSs 1 and 2, tries to switch her RRC to a lower RRC managed by cell 1. Also, let $\alpha(k)$ and $\alpha'(k)$ be the resource ratios before and after the switch, respectively. In order to determine the user's throughput post-switching for a given $\alpha'(k)$, the system can use a capacity $C_1(k)$ which is computed using the average SINR reported by user k had she been an interior user in cell 1.

As in the case of upward RRC switch, user k will accept the RRC switch when her throughput becomes higher by changing the RRC. Consequently, the first condition for the downward RRC switch is given by

$$\alpha'(k)C_1(k) - \alpha(k)C_{1,2}(k) \geq 0. \tag{16}$$

Next, the impact of the downward RRC switch on the system throughput depends on the following factors:

- *The throughput gain in cell 2:* The reclassification of user k will release a ratio $\alpha(k)$ of resource in BS 2 which can be distributed to the cell-interior users in BS 2. Thus, the average sum throughput gain by interior users in cell 2 can be quantified as $\alpha(k) \cdot \bar{C}_2$.
- *The throughput change in cell 1:* Notice that if $\alpha'(k) < \alpha(k)$, the residual part $\alpha(k) - \alpha'(k)$ can be distributed to cell 1's interior users who will together achieve an average throughput gain of $(\alpha(k) - \alpha'(k)) \cdot \bar{C}_1$. Otherwise, i.e., if $\alpha'(k) > \alpha(k)$, they will lose an average throughput of $(\alpha'(k) - \alpha(k)) \cdot \bar{C}_1$.
- *System Constraints:* In the case $\alpha'(k) > \alpha(k)$, the additional resource $\alpha'(k) - \alpha(k)$ taken from BS 1 should not jeopardize the minimum throughput requirement of any of its interior users in L_1 .

Therefore, a post-switching ratio $\alpha'(k)$ is acceptable only if it leads to an increase in system throughput, i.e., it satisfies

$$\alpha'(k)[C_1(k) - \bar{C}_1] - \alpha(k)[C_{1,2}(k) - \bar{C}_1 - \bar{C}_2] \geq 0, \quad (17)$$

and also respects the system constraints, i.e.,

$$\alpha'(k) \leq 1 - \beta_1 - \sum_{i \in L_1} \frac{T_{\min}(i)}{C_x(i)}. \quad (18)$$

Thus, the optimal ratio $\alpha'(k)$ can be determined by solving the following optimization problem:

$$\begin{aligned} \max \quad & \alpha'(k)[C_1(k) - \bar{C}_1] - \alpha(k)[C_{1,2}(k) - \bar{C}_1 - \bar{C}_2] \\ \text{subject to} \quad & \text{(16), (17), and (18)} \end{aligned} \quad (19)$$

The solution to the above problem is given by the following proposition. The proof is omitted because it is similar to that of the previous proposition corresponding to the upward RRC switch.

Proposition 2. *The conditions for reclassifying a user k and changing her RRC from an upper RRC to a lower RRC is summarized as follows.*

i) If $C_1(k) - \bar{C}_1 < 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 < 0$, then switching is allowed only if

$$(\bar{C}_1 + \bar{C}_2) \cdot C_1(k) - \bar{C}_1 \cdot C_{1,2}(k) > 0, \quad (20)$$

and if the post-switching ratio $\alpha(k)C_{1,2}(k)/C_1(k)$ meets the condition (18). The optimal $\alpha'(k)$, when these two conditions are met, is given by

$$\alpha'(k) = \alpha(k)C_{1,2}(k)/C_1(k). \quad (21)$$

ii) If $C_1(k) - \bar{C}_1 = 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 < 0$, $\alpha'(k)$ can be chosen arbitrarily subject to (16) and (18).

iii) If $C_1(k) - \bar{C}_1 > 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 \leq 0$, $\alpha'(k)$ should be the maximal possible value subject to (16) and (18).

Remark 1. The case of $C_1(k) - \bar{C}_1 > 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 > 0$ can be considered in both upward and downward switchings, where $\alpha'(k)$ should be the maximal possible value subject to other constraints. Suppose user k in cell 1 is served by a lower RRC and satisfies $C_1(k) - \bar{C}_1 > 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 > 0$. Then, if upward switching is permitted, the user will seek a maximal $\alpha'(k)$ subject to the other conditions required for the upward RRC switch. Upon switching, the user will then try to switch to a lower RRC, again seeking a maximal $\alpha'(k)$ subject to the other conditions required for the downward RRC switch. It can be verified that an upward (third) switch will be not possible and the same observation holds if the user were originally served by the upper RRC. Thus, users satisfying $C_1(k) - \bar{C}_1 > 0$ and $C_{1,2}(k) - \bar{C}_1 - \bar{C}_2 > 0$ may switch at most twice, and in our simulation, such users are observed to mainly remain in the lower RRC.

Remark 2. We now justify the extra condition we imposed that a user’s throughput must not decrease upon switching, instead of just requiring an increase in system throughput for switching, where the latter will be referred to as relaxed switching in the sequel. This additional constraint ensures better cell-edge performance by protecting cell edge users against loss in throughput. Consider the upward switch of a user in cell 1 and assume that an upward switch is possible in relaxed upward switching but not in our switching. In this case, with upward relaxed switching, the system can decide to reclassify an interior user with a lower average capacity as a cell-edge user and allocate a resource ratio just enough to meet its minimum throughput. Moreover, the increase in system throughput in this case is due to an increase in the sum throughput of cell 1’s other interior users. A similar observation holds for the downward switch case. Thus the additional constraint prevents the system from using switching to boost system throughput by starving edge users.

3.3 Simplified Solutions

We now develop simplified solutions for both the RRC switch algorithms when the capacity of an interior user can be computed using eq. (11). We make the assumption that in order to be eligible for switching a user must satisfy $C_1(k) < \overline{C}_1$ as well as $C_{1,2}(k) < \overline{C}_1 + \overline{C}_2$. Note that this assumption is reasonable since the average capacity of a user k at the edge of cell 1 will be smaller than the average per-user capacity of the cell-interior users and is validated in our simulation. As a consequence, only the first cases in both *Propositions 1* and *2* are now possible and we address them below.

Fractional frequency reuse. Suppose fractional frequency reuse is employed to support the cell-edge users. Now consider the upward RRC switch. Using the capacity expression given in eq. (3), the condition in (14) can be expressed as

$$\overline{C}_1 \log(1 + \gamma_1) - (\overline{C}_1 + \overline{C}_2) \log\left(1 + \frac{\gamma_1}{\gamma_2 + 1}\right) \geq 0. \tag{22}$$

The above expression in turn can be compactly written as

$$(1 + \gamma_1)(1 + \gamma_2)^{1+\lambda} \geq (1 + \gamma_1 + \gamma_2)^{1+\lambda}, \tag{23}$$

where $\lambda = \overline{C}_2/\overline{C}_1$. Similarly, it can be shown that the corresponding condition for the downward RRC switch is given by (23) but where the inequality is reversed.

Macro-diversity. Next, when macro-diversity is employed to support the cell-edge users, using the capacity expression given in eq. (5), the condition (14) in the upward RRC switch can be expressed as

$$\overline{C}_1 \log(1 + \gamma_1 + \gamma_2) - (\overline{C}_1 + \overline{C}_2) \log\left(1 + \frac{\gamma_1}{\gamma_2 + 1}\right) \geq 0. \tag{24}$$

This can be further rewritten as

$$(1 + \gamma_2)^{1+\lambda} \geq (1 + \gamma_1 + \gamma_2)^\lambda. \quad (25)$$

The corresponding condition for the downward RRC switch is given by (25) but where the inequality is reversed.

Therefore, in order to decide the RRC switch using the simplified conditions, each user can report γ_1 and γ_2 to the admission controller, and the admission controller should be able to determine λ . The role of γ_1 , γ_2 and λ is highlighted in the following proposition.

Proposition 3. *An upward RRC switch requires an increasing value of γ_2 as λ increases, given an arbitrarily fixed γ_1 . Conversely, a downward RRC switch requires a decreasing value of γ_1 as λ increases, given an arbitrarily fixed γ_2 .*

The proof is given in Appendix B.

Fig. 1 depicts the boundary conditions of switching a RRC as a function of γ_1 and γ_2 as given by (23) and (25) for fractional frequency reuse and macro-diversity, respectively. As stated in Proposition 3, a greater γ_2 is needed for an upward switch when λ is higher.

Thus far, we have assumed that the average capacity of the interior users in a neighboring cell is available. When this information is unavailable in the network, or each user (instead of an admission controller) independently wants to decide the RRC switch without network-level information, we can obtain approximate conditions assuming $\lambda = 1$ (i.e., $\overline{C}_1 = \overline{C}_2$). Then the conditions of (14) and (20) are simply expressed by

$$C_{1,2}(k) - 2C_1(k) \geq 0 \text{ and } C_{1,2}(k) - 2C_1(k) < 0. \quad (26)$$

Specifically, in the case of macro-diversity, the boundary condition in (25) is given by

$$\gamma_2 = \frac{(1 + 4\gamma_1)^{1/2} - 1}{2}, \quad (27)$$

which provides an insight for designing *H_Add Threshold* and *H_Delete Threshold* for *macro diversity hand-off procedure* defined in the IEEE 802.16e standard [11].⁶

3.4 Overhead of RRC Switch

Throughout this paper, we consider stationary (fixed or nomadic) users. Fast-moving users can always be managed by the upper RRC regardless of whether

⁶ The IEEE 802.16e standard introduces a macro diversity hand-off procedure where a mobile user is able to transmit or receive unicast messages and traffic from multiple BSs at the same time interval. According to [11], when the long-term SINR of a serving BS is less than *H_Delete Threshold*, the mobile station shall send MOB_MSHO-REQ to require dropping this serving BS from the diversity set, and when the long-term SINR of a neighboring BS is higher than *H_Add Threshold*, the mobile station shall send MOB_MSHO-REQ to require adding this neighbor BS to the diversity set.

they are classified as cell-interior or cell-edge users, but we do not consider such mobility issues here. Stationary users compute γ_1 and γ_2 via a long-term average, so most users will not suffer from frequent RRC switch. The overhead of RRC switch is the exchange of signaling messages for switch request and response between two RRCs. If the algorithms are triggered more frequently, the classification will probably be more accurate, but the overhead will be higher.

4 Simulation Results

We evaluated the performance in an OFDMA-based wireless network by simulation experiments, emulating mobile WiMAX systems with parameters listed in Table 1. We consider a single omnidirectional antenna at each transmitter and each receiver. In our simulator, users are uniformly distributed in a hexagonal cell and BSs of 6 first-tier and 12 second-tier neighboring cells generate intercell interference to those users. Our channel model follows path loss with an exponent of 4, Gaussian shadowing with zero mean and variance of 8 dB, and Rayleigh fading. We use the Jakes' model [13] to generate frequency-selective Rayleigh fading followed by the Doppler effect with the maximum velocity of 3 Km/hr. To serve cell-interior users, BSs either adopt a round-robin (RR) scheduling algorithm or a multi-channel proportional fair (PF) scheduling algorithm [14] that guarantees minimum throughput (150 Kbps for all users in our setting) [10]. It is assumed that the channel coefficients are perfectly known at the BS and the data rate is then determined by the Shannon capacity. In our simulation, each user measures the two strongest γ 's from her neighboring BSs, and the serving BS is able to coordinate with one or two of those neighboring BSs. The cell performance was computed during the simulation time of 60 seconds, after each user's RRC had been completely determined according to our algorithm.

Our simulation results show that users are appropriately classified into cell-edge and cell-interior types by our algorithms. We confirmed that i) the first cases in Propositions 1 and 2 are generally observed, ii) FFR and macro-diversity (MD) increase cell-edge throughput by up to 15% when $\lambda = 1$ without a loss in system throughput, and iii) more users switch to the cell-edge type when the neighboring cell is lightly loaded.

First, Fig. 2 shows the distribution of cell-edge users' γ_1 and γ_2 in the case of Fig. 1, when macro-diversity by at most two BSs is employed and $\bar{C}_1 = \bar{C}_2$. The black area represents γ_1 and γ_2 of those users by simulation results and two lines

Table 1. Parameters for simulation [12]

Channel bandwidth	5 MHz	No. of sub-channels	8
Carrier frequency	2.3 GHz	TX power at BSs	43 dBm
Cell radius	1 Km	Path loss exp.	4
Shadowing var.	8 dB	Max. Doppler vel.	3 Km/hr
Number of users	30	$T_{\min}(i)$	150 Kbps
Simulation time	60 seconds	No. of simulations	1000

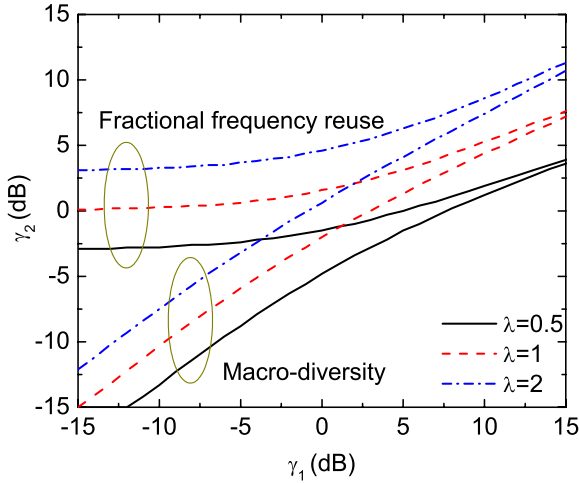


Fig. 1. Boundary conditions of switching a RRC

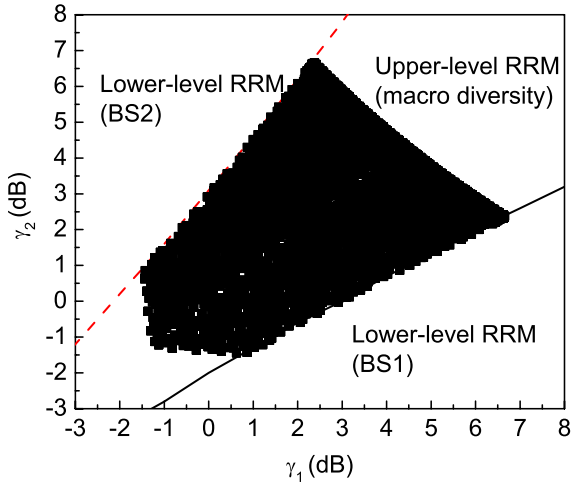


Fig. 2. Distribution of cell-edge users' γ_1 and γ_2 when macro-diversity is used and $\overline{C}_1 = \overline{C}_2$

represent the threshold given by (14). In this experiment, the other cases except the first one in Propositions 7 and 8 are rarely observed; for instance, the ratio of such cases is only 0.5% among all users at $\lambda = 0.2$ and it approaches zero as

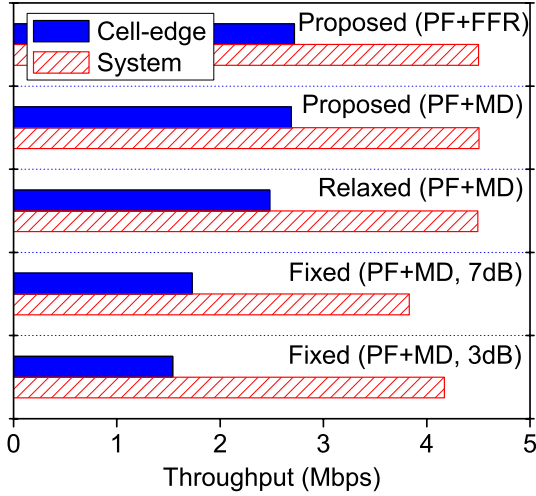


Fig. 3. Comparison of cell-edge users’ average throughput and system throughput in various mechanisms

λ increases above 0.2. Therefore, as expected, the first cases can be regarded as the simplified solution in general.

The average cell-edge throughput and system throughput (i.e., cell throughput in this simulation) are presented in Fig. 3 when $\lambda = 1$. Both “PF+FFR” and “PF+MD” represent the cases where cell-interior users are supported by the PF scheduling and cell-edge users are supported by FFR or MD. The proposed algorithm shows better cell-edge throughput, compared to the relaxed switching (“Relaxed”) mentioned in Remark 2. Compared to the case of no upper RRC, the proposed one improves cell-edge throughput by 13.0% and 14.3% for FFR and MD, respectively, without a loss in system throughput, while the relaxed case improves it only by 4.2%. Also, our algorithm is compared to a simple mechanism (represented by “Fixed”) where RRC switch is determined by a fixed threshold, $\gamma_1 - \gamma_2$ (3dB or 7dB). Here γ_2 is given by the neighboring BS that interferes most dominantly. In summary, the proposed algorithm achieves the best cell-edge throughput without losing system throughput. We omit “PF+FFR” for the fixed and relaxed switching because it results in a slightly inferior cell-edge performance to “PF+MD”.

The effect of λ is demonstrated in Fig. 4 that plots β as a function of λ . Here, β also includes the fraction of resource allocated to cell-edge users who are located in six neighboring cells. As discussed in Proposition 3, users are less likely to switch to the upper RRC as λ increases. To obtain this result, we imposed no upper limit on β (i.e., $\beta_{\max} = 1$). When RR scheduling is employed for cell-interior users, they do not take advantage of opportunistic scheduling,

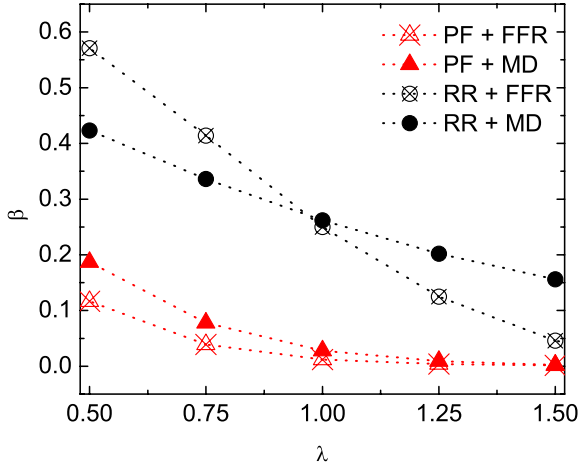


Fig. 4. β vs. λ

and thus it drives more users to switch to the upper RRC. Therefore, β in case of RR scheduling is much greater than that of PF scheduling.

5 Conclusion

We have proposed a new RRM framework for wide-area wireless data networks that manages radio resources of cell-interior and cell-edge users separately. The work presented in this paper has been limited to downlink data transmission; RRM schemes for uplink in conjunction with downlink would be one avenue for future work.

Acknowledgement

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0003).

References

1. Li, G., Lu, H.: Downlink Radio Resource Allocation for Multi-cell OFDMA System. IEEE Trans. Wireless Commun. 5(12), 3451–3459 (2006)
2. Elayoubi, S.-E., Ben Haddada, O., Fourestie, B.: Performance Evaluation of Frequency Planning Schemes in OFDMA-based Networks. IEEE Trans. Wireless Commun. 7(5) (May 2008)

3. Qualcomm R1-050896, Description and Simulations of Interference Management Technique for OFDMA based E-UTRA Downlink Evaluation, 3GPP TSG-RAN WG1 #42 (August 2005)
4. Kim, S., Kim, J., Lim, D., Ihm, B.-C., Cho, H.: Interference Mitigation Using FFR and Multi-Cell MIMO in Downlink. IEEE C802.16m-08/783r1 (July 2008)
5. Bernhardt, R.: Macroscopic Diversity in Frequency Reuse Radio Systems. IEEE J. Select. Areas Commun. 5(5), 862–870 (1987)
6. Caire, G., Muller, R.R., Knopp, R.: Hard Fairness Versus Proportional Fairness in Wireless Communications: The Single-Cell Case. IEEE Trans. Inform. Theory 53(4), 1366–1385 (2007)
7. Choi, J.-G., Bahk, S.: Cell-Throughput Analysis of the Proportional Fair Scheduler in the Single-Cell Environment. IEEE Trans. Vehi. Tech. 56(2), 766–778 (2007)
8. Alamouti, S.M.: A Simple Transmit Diversity Technique for Wireless Communications. IEEE J. Sel. Areas Commun. 16(8) (October 1998)
9. Tarokh, V., Jafarkhani, H., Calderbank, A.R.: Space-Time Block Codes from Orthogonal Designs. IEEE Trans. Inform. Theory 45(5), 744–765 (1999)
10. Andrews, M., Qian, L., Stolyar, A.: Optimal Utility based Multi-user Throughput Allocation Subject to Throughput Constraints. In: Proc. IEEE INFOCOM 2005, Miami, FL, USA (March 2005)
11. IEEE 802.16e-2005, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment (February 2006)
12. WiMAX forum, Mobile WiMAX - Part I: A Technical Overview and Performance Evaluation (August 2006)
13. Jakes, W.C.: Microwave Mobile Communications. John Wiley & Sons, Chichester (1975)
14. Kim, H., Han, Y.: A Proportional Fair Scheduling for Multicarrier Transmission Systems. IEEE Commun. Lett. 9(3), 210–212 (2005)

A Appendix: Proof of Proposition 1

In the case of i), the RRC switch is possible if an $\alpha'(k)$ exists such that

$$\alpha(k) \frac{C_1(k)}{C_{1,2}(k)} \leq \alpha'(k) \leq \alpha(k) \frac{\overline{C}_1 - C_1(k)}{\overline{C}_1 + \overline{C}_2 - C_{1,2}(k)}, \quad (28)$$

which is obtained from (10) and (11). The upper bound must be greater than the lower bound, which results in (14). The objective is maximized by the minimal value, i.e., $\alpha'(k) = \alpha(k)C_1(k)/C_{1,2}(k)$. The proofs of the other cases are omitted because they follow along similar lines.

B Appendix: Proof of Proposition 3

For brevity, we only prove the case of upward switching. In the case of fractional frequency reuse, (23) is equivalent to

$$\lambda < \frac{\log(1 + \gamma_1)}{\log(1 + \gamma_1/(\gamma_2 + 1))} - 1 \triangleq f(\gamma_2) \quad (29)$$

In the case of macro-diversity, (25) can be re-written as

$$\lambda < \frac{1}{1 - \frac{\log(1+\gamma_2)}{\log(1+\gamma_1+\gamma_2)}} - 1 \triangleq g(\gamma_2) \quad (30)$$

It is easily proved that for a fixed γ_1 , $f(\gamma_2)$ and $g(\gamma_2)$ are monotonically increasing functions of γ_2 . Therefore, as the average capacity of a neighboring cell 2 increases (i.e. as λ increases), an increasing value of γ_2 is required.

Malicious or Selfish? Analysis of Carrier Sense Misbehavior in IEEE 802.11 WLAN

Kyung-Joon Park¹, Jihyuk Choi², Kyungtae Kang¹, and Yih-Chun Hu²

¹ Department of Computer Science,
University of Illinois at Urbana-Champaign, 201 N. Goodwin Avenue, Urbana, IL 61801 USA
{kjjo, ktakang}@illinois.edu

² Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign, 1406 West Green Street, Urbana, IL 61801 USA
{jchoi43, yihchun}@illinois.edu

Abstract. The behavior of selfish users, which does not respect the backoff procedure of IEEE 802.11 WLAN, has been nicely studied in game-theoretic frameworks. However, in these studies, the effect of physical carrier sense has not been properly incorporated into the analysis. In this paper, we study the problem of how carrier sense misbehavior can affect network performance in addition to backoff misbehavior. Our analysis shows that a cheater can increase its throughput by ignoring the carrier sense mechanism while a well-behaved user significantly loses its throughput. Consequently, not only a malicious user, but also a selfish one is motivated to disregard the carrier sense mechanism, which will result in significant throughput degradation of well-behaved ones. Our analysis also shows that carrier sense misbehavior corresponds to the case of virtually increasing the channel access probability of the cheater in the backoff procedure. We provide our preliminary simulation results, which verify our analysis.

Keywords: IEEE 802.11 MAC, physical carrier sense, wireless network security, MAC-layer misbehavior.

1 Introduction

With the ubiquitous deployment of wireless networks such as IEEE 802.11 Wireless Local Area Network (WLAN), it becomes critical to protect the network from malicious and selfish users. In particular, the Medium Access Control (MAC) protocols in wireless networks typically adopt distributed contention resolution schemes for the shared wireless channel. Hence, a misbehaving user that does not respect MAC protocols can significantly degrade network performance by diminishing the bandwidth share of well-behaved users. For example, the IEEE 802.11 WLAN MAC adopts the Distributed Coordination Function (DCF), which basically consists of the CSMA/CA with the exponential backoff mechanism. Consequently, a malicious or selfish user, which disregards the IEEE 802.11 DCF, may cause significant performance degradation of well-behaved ones.

Recently, selfish or greedy behavior in IEEE 802.11 MAC has been substantially studied [1, 2, 3, 4, 5, 6]. In particular, the selfish behavior disregarding the exponential

backoff mechanism has been nicely formulated in game-theoretic frameworks [2, 3, 5]. These studies have shown that the selfish behavior of disregarding the exponential backoff mechanism can significantly increase the bandwidth share of a cheater at the expense of that of a normal user. In addition, several efficient detection mechanism for misbehavior of a cheater have been proposed [2, 4, 5]. However, in these studies, the effect of physical carrier sense has not been considered in the analysis.

In this paper, we study the problem of how carrier sense misbehavior affects network performance in IEEE 802.11 WLAN. Since the basic IEEE 802.11 MAC consists of physical carrier sense and the exponential backoff mechanism, carrier sense misbehavior provides another line of possibilities for a cheater in addition to disregarding the exponential backoff mechanism. We consider the case when a cheater does not respect the carrier sense mechanism with a certain cheating rate when it has a packet to send. We first derive the saturation throughput of a cheater and a well-behaved user as a function of the cheating rate, respectively. Then, we show that the throughput of a cheater increases while that of a normal user decreases as the cheating rate increases. Consequently, similarly as in the case of the backoff misbehavior, the cheater can increase its bandwidth share by sacrificing that of the normal user. Our analysis further shows that carrier sense misbehavior corresponds to the case of virtually increasing the channel access probability of the cheater in the exponential backoff mechanism. Our simulation results validate our analysis.

The remainder of the paper is organized as follows. We introduce recent studies on wireless network security in Section 2. Then, in Section 3, we introduce the IEEE 802.11 DCF and the Bianchi's model for the exponential backoff procedure, which will be used in our analysis in subsequent sections. In Section 4, we derive the saturation throughput of a cheater and a well-behaved user as a function of the cheating rate, respectively. Then, we show that the throughput of a cheater increases while that of a normal user decreases as the cheating rate increases. Simulation results that verify our analysis are given in Section 5. Finally, our conclusion follows in Section 6.

2 Related Work

The IEEE 802.11 standard has seen successful widespread deployment because of its unlicensed spectrum and low hardware cost. The original security protocol of IEEE 802.11, called Wired Equivalent Privacy (WEP), was designed to provide privacy and authenticity of data. However, it has been shown by Fluhrer et al. [7] that weakness in the encryption algorithm used by WEP can be exploited to allow the discovery of session keys. After this study, various related attacks have been demonstrated, for example, [8, 9].

Bellardo and Savage [10] have implemented and demonstrated an attack that targets the authentication/association scheme of IEEE 802.11. They showed that the deauthentication and disassociation messages are not encrypted in the scheme, and thus an attacker can easily forge these messages. The attacker can then send the deauthentication message to the access point before client's data is received, or the attacker can send the disassociation message to the client before the client's data is transmitted. They further showed in [10] that the 802.11 carrier sense mechanism can be easily exploited. For

example, in 802.11 networks, a node can only send data a certain time after the channel stops being busy. In particular, if not due to retransmission or fragmentation, a user can only transmit data DCF InterFrame Space (DIFS) after channel is available; otherwise the user can transmit data Short InterFrame Space (SIFS) after, where $SIFS < DIFS$. Bellardo and Savage then presented a sophisticated scheme exploiting the virtual carrier sense mechanism. The 802.11 standard specifies that the MAC frame header of all packets should contain a *duration* field, which specifies how long others have to wait before transmission is allowed in order to avoid collision. Users update their Network Allocation Vector (NAV) with this duration information and keep quiet for the specified duration. Thus an attacker can repeatedly request long channel occupancy time, thereby starving normal clients of channel occupancy.

Another type of attack that disregards the backoff procedure in 802.11 MAC has been substantially studied [1][2][3]. In these studies, game-theoretic frameworks have often played a key role for analyzing the network behavior [2][3]. For example, it has been shown in [2] that the backoff misbehavior leads to a significant unfair share of bandwidth between the cheater and the well-behaved users. Then, an efficient game-theoretic framework has been proposed to drive the network operating point to a pareto-optimal one. More recently, Pelechrinis et al. [6] showed in their empirical studies that carrier sense misbehavior significantly degrades the performance of well-behaved users while the cheater can substantially increase its bandwidth share. Based on this observation, They proposed a scheme for detecting this misbehavior in IEEE 802.11 WLAN. Their key idea is as follows: Since the cheater will ignore low-power receptions as legitimate packets, by intelligently sending low-power probe packets, an AP can detect the cheater with high probability.

Our study here lies in the direction of these studies on carrier sense misbehavior. Our main focus is on investigating the performance of each user with carrier sense misbehavior, which has not been considered in the analysis of previous studies. Though the throughput performance with carrier sense misbehavior has been empirically shown in [6], there has been few analytical studies on this issue. Consequently, we look into this problem in an analytical manner, which we expect will be a building block for developing efficient detection and prevention mechanisms for carrier sense misbehavior in the future.

3 Preliminaries

3.1 IEEE 802.11 DCF Mechanism

The basic CSMA/CA mechanism in IEEE 802.11 DCF operates as follows. When a station has a frame to transmit, it senses the medium first, which is called *physical carrier sense*. After the medium is sensed idle for a time interval of Distributed InterFrame Space (DIFS), it starts to transmit the frame. Otherwise, the station defers its transmission according to an exponential backoff algorithm: It maintains a random backoff timer, whose value is uniformly distributed in $[0, CW]$, where CW stands for the contention window size. CW is always 1 less than a power of 2 (e.g., 15, 31, 63, ...). CW is initially set to its minimum value of CW_{min} , moves to the next greatest power of two, up to its maximum value of CW_{max} , after each time the frame incurs a collision. The backoff

timer is decremented by one for each physical slot time σ when the channel is idle, suspended whenever the channel is busy, and reactivated after the channel is sensed idle again for a DIFS. The node transmits when the backoff timer reaches zero. After transmitting frame except broadcasting message, the source node expects to receive a positive acknowledgement (ACK) frame from the destination node within an interval of Short InterFrame Space (SIFS). If an ACK is not received in SIFS, the sender assumes the frame has experienced a collision, and schedules a retransmission for this frame while updating CW according to the exponential backoff algorithm.

3.2 Markov Chain Model for the IEEE 802.11 Exponential Backoff Mechanism

Here, we briefly introduce the Markov chain model for the IEEE 802.11 exponential backoff mechanism in [11] for completeness, which will be used in our analysis in the next section. For a given node, each state is represented as (i, k) where i is the backoff stage and k is the current backoff counter. The backoff window size at stage i is denoted by W_i . Since the minimum backoff window size is W , W_i becomes $W_i = 2^i W$ with the binary exponential backoff¹. The maximum backoff stage is m . Then, the one-step transition probabilities are as follows:

$$\begin{cases} P\{i, k \mid i, k + 1\} = 1, & k \in (0, W_i - 2), i \in (0, m); \\ P\{0, k \mid i, 0\} = \frac{(1-p)}{W_0}, & k \in (0, W_0 - 1), i \in (0, m); \\ P\{i, k \mid i - 1, 0\} = \frac{p}{W_i}, & k \in (0, W_i - 1), i \in (i, m); \\ P\{m, k \mid m, 0\} = \frac{p}{W_m}, & k \in (0, W_m - 1). \end{cases}$$

Now, let $b_{i,k}$ denote the stationary probability of state (i, k) . Then, we have the following relation.

$$\begin{cases} b_{i,0} = p^i b_{0,0}, & 0 < i < m; \\ b_{m,0} = \frac{p^m b_{0,0}}{(1-p)}. \end{cases}$$

Then, we have

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0}, \quad i \in (0, m), k \in (1, W_i - 1).$$

Finally, $b_{i,k}$ can be expressed as a function of $b_{0,0}$ as follows:

$$\begin{cases} b_{i,k} = \frac{p^i (W_i - k)}{W_i} b_{0,0}, & i \in (0, m - 1), k \in (1, W_i - 1); \\ b_{m,k} = \frac{p^m (W_m - k)}{(1-p) W_m} b_{0,0}, & k \in (1, W_m). \end{cases} \quad (1)$$

Now, $b_{0,0}$ can be obtained by applying the normalization condition of the Markov chain as follows:

$$1 = \sum_{i=0}^m \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^m b_{i,0} + \sum_{i=0}^m \sum_{k=1}^{W_i-1} b_{i,k}. \quad (2)$$

¹ In fact, W_i and W correspond to $CW + 1$ at stage i and $CW_{min} + 1$ in the previous section, respectively.

By using (1), (2) gives

$$b_{0,0} = \frac{2(1 - 2p)(1 - p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)}. \tag{3}$$

From (3), the channel access probability τ can be obtained as follows.

$$\tau = \sum_{i=0}^m b_{i,0} = \frac{2(1 - 2p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)}. \tag{4}$$

4 Performance Analysis with Carrier Sense Misbehavior

In this section, we present an analytical framework for modeling a heterogeneous IEEE 802.11 WLAN, where a cheater and a well-behaved user coexist. In our analysis, we assume that there exist one misbehaving user and one normal user in the network. A more general analysis will be an issue of future work.

4.1 System Descriptions and Assumptions

There are basically two ways for a cheater to disregard the carrier sense mechanism. First, it can intentionally ignore the ongoing transmission of a well-behaved user during the exponential backoff procedure and attempt to transmit by decrementing its backoff counter without freezing. In this case, if the transmission of a frame takes longer than a usual backoff window, it is clear that both of the transmissions will fail because of the collision, which is apparently malicious to both the cheater and the well-behaved user. Another way is to disregard the carrier sense mechanism at the beginning, and starts to transmit without entering into the exponential backoff mechanism. Here, we consider the latter case, under which the cheater may benefit from its misbehavior.

Without loss of generality, let User 1 denote the cheater and User 2 the well-behaved one. The conditional collision probability and the channel access probability of each user are denoted by p_i and τ_i , $i = 1, 2$, respectively. In addition, the cheater ignores the carrier sense mechanism with a cheating rate of q , i.e., with a probability of q , the cheater accesses the channel without carrier sense (which also results in bypassing the exponential backoff mechanism). Consequently, the cheater can affect the throughput performance of both users by adjusting the value of q . Our analysis has the following two goals; to derive the throughput of each user as a function of the cheating rate q and to identify the effect of q on the throughput of each user based on the derived model.

Note that it is not a simple task to discover the analytical relation between the throughput of each user and the cheating rate because of the exponential backoff procedure. In addition, it should be noted that the channel access probability of the cheater, τ_1 , is defined as the conditional channel access probability when the cheater has decided not to cheat (which occurs with a probability of $(1 - q)$). The actual channel access probability of the cheater seen by the well-behaved one is different from τ_1 . We will discuss this issue in the subsequent section.

4.2 Markov Chain Model for the Exponential Backoff Mechanism with Carrier Sense Misbehavior

Here, by using (4), we derive the systems of equations for the exponential backoff mechanism with carrier sense misbehavior. Note that our analysis is different from the homogeneous case in (11) in the sense that we consider the heterogeneous situation where τ_i 's and p_i 's are different, respectively, because of the introduction of the carrier sense cheating rate q .

As already introduced in the previous sections, let p_i and τ_i , $i = 1, 2$ denote the collision probability and the channel access probability of the cheater and the well-behaved user, respectively. Since the event of cheating is independent of the well-behaved user's channel access, the conditional collision probability of the cheater, p_1 , becomes

$$p_1 = 1 - (1 - \tau_2) = \tau_2. \tag{5}$$

In the meantime, the channel access of the well-behaved user will succeed if the cheater has decided not to cheat and further decided not to transmit according to the exponential backoff mechanism. Hence, p_2 becomes

$$p_2 = 1 - (1 - q)(1 - \tau_1) = 1 - [1 - \{(1 - q)\tau_1 + q\}] = \tau'_1, \tag{6}$$

where $\tau'_1 = (1 - q)\tau_1 + q$. In addition, from (4), we have

$$\tau_i = \frac{2(1 - 2p_i)}{(1 - 2p_i)(W + 1) + p_iW(1 - (2p_i)^m)}, i = 1, 2. \tag{7}$$

Hence, from (5), (6), and (7), we have

$$\tau_1 = F(\tau_2) = \frac{2(1 - 2\tau_2)}{(1 - 2\tau_2)(W + 1) + \tau_2W(1 - (2\tau_2)^m)}, \tag{8}$$

and

$$\tau_2 = G(\tau_1, q) = \frac{2(1 - 2\tau'_1)}{(1 - 2\tau'_1)(W + 1) + \tau'_1W(1 - (2\tau'_1)^m)}, \tag{9}$$

where $\tau'_1 = (1 - q)\tau_1 + q$. Similarly as in the homogeneous case in (11), (8) and (9) constitutes a nonlinear system of equations for τ_1 and τ_2 . It should be noted in (9) that the access probability of the well-behaved user, τ_2 , is determined by the exponential backoff procedure in (4) as if the cheater accesses the channel with $\tau'_1 = (1 - q)\tau_1 + q$. We have the following relation for F in (8):

Lemma 1. $F(\tau_2)$ in (8) is a decreasing function of τ_2 .

Proof. From (8), we can easily show that $dF(\tau_2)/d\tau_2 < 0$. □

In addition, we have the following result for G in (9):

Lemma 2. For a given value of τ_1 , $G(\tau_1, q)$ is a decreasing function of q .

Proof. It is straightforward from (9) that $\partial G(\tau_1, q)/\partial q < 0$. □

From Lemma 1 and Lemma 2, the solution to the systems of equations in (8) and (9) has the following property:

Theorem 1. *Let (τ_1^*, τ_2^*) denote the solution to the system of (8) and (9). Then, (τ_1^*, τ_2^*) is unique. Furthermore, τ_1^* increases with q while τ_2^* decreases with q .*

Proof. From (8), $F(0) = 2/(W + 1)$ and $F(1) = 2/(2^m W + 1)$. Similarly, from (9), we have $G(0, q) = 2(1 - 2q)/[(1 - 2q)(W + 1) + qW(1 - (2q)^m)] = F(q)$ and $G(1, q) = 2/(2^m W + 1) = F(1)$. Since $0 \leq q \leq 1$ and F is a decreasing function from Lemma 1, we have $F(q) \geq F(1)$. Hence, it can be concluded that (τ_1^*, τ_2^*) is unique. Furthermore, since G is a decreasing function of q from Lemma 2, τ_1^* is an increasing function of q while τ_2^* is a decreasing function of q . \square

From Theorem 1, we have the following corollary:

Corollary 1. *The virtual access probability of the cheater, denoted by $\tau_1' = (1 - q)\tau_1 + q$, is an increasing function of q .*

Proof. By differentiating τ_1' with respect to q , we have

$$\frac{\partial \tau_1'}{\partial q} = (1 - \tau_1) + (1 - q) \frac{\partial \tau_1}{\partial q}.$$

Since $0 \leq q, \tau_1 \leq 1$ and $\partial \tau_1 / \partial q > 0$ from Theorem 1, we have $\partial \tau_1' / \partial q > 0$. \square

Remark 1. *Our analysis shows that the effect of carrier sense misbehavior is to virtually increase the channel access probability of the cheater seen by the normal user from τ_1 to $\tau_1' = (1 - q)\tau_1 + q$. From Corollary 1, the virtual channel access probability of the cheater increases as the cheating rate q increases. It should be noted that τ_1 is still determined according to the ordinary exponential backoff mechanism as given in (8).*

4.3 Saturation Throughput of Heterogeneous IEEE 802.11 WLAN with Carrier Sense Misbehavior

Let $v_i, i = 1, 2$ denote the virtual slot time, which is the average time for each event of User i . Then, we have

$$\begin{aligned} v_1 = v_2 &= (1 - q)[(1 - \tau_1)(1 - \tau_2)\sigma + \{\tau_1(1 - \tau_2) + \tau_2(1 - \tau_1)\}T_s + \tau_1\tau_2T_c] \\ &\quad + q\{(1 - \tau_2)T_s + \tau_2T_c\} \\ &= (1 - \tau_1')(1 - \tau_2)\sigma + \{\tau_1'(1 - \tau_2) + \tau_2(1 - \tau_1')\}T_s + \tau_1'\tau_2T_c, \end{aligned} \tag{10}$$

where σ, T_s , and T_c denote the slot time, the time for successful transmission, and that for collision, respectively.

Let $S_i, i = 1, 2$ denote the saturation throughput of User i . Then, from (5), we have

$$S_1 = \frac{((1 - q)\tau_1 + q)(1 - p_1)}{v_1} = \frac{\tau_1'(1 - \tau_2)}{v_1},$$

where $\tau_1' = (1 - q)\tau_1 + q$ and v_1 is given in (10). We have the following relation between S_1 and q :

Theorem 2. *The saturation throughput of the cheater is an increasing function of the cheating rate q .*

Proof. By applying the chain rule,

$$\begin{aligned} \frac{\partial S_1}{\partial q} &= \frac{\partial S_1}{\partial \tau_1'} \frac{\partial \tau_1'}{\partial q} + \frac{\partial S_1}{\partial \tau_2} \frac{\partial \tau_2}{\partial q} + \frac{\partial S_1}{\partial v_1} \frac{\partial v_1}{\partial q} \\ &= \frac{(1 - \tau_2)}{v_1} \frac{\partial \tau_1'}{\partial q} - \frac{\tau_1'}{v_1} \frac{\partial \tau_2}{\partial q} - \frac{\tau_1'(1 - \tau_2)}{v_1^2} \frac{\partial v_1}{\partial q}. \end{aligned} \tag{11}$$

In the meantime, from (10),

$$\frac{\partial v_1}{\partial q} = \frac{\partial v_1}{\partial \tau_1'} \frac{\partial \tau_1'}{\partial q} + \frac{\partial v_1}{\partial \tau_2} \frac{\partial \tau_2}{\partial q}. \tag{12}$$

By plugging (12) into (11), we have

$$\frac{\partial S_1}{\partial q} = \frac{(1 - \tau_2)}{v_1} \left[1 - \frac{\tau_1'}{v_1} \frac{\partial v_1}{\partial \tau_1'} \right] \frac{\partial \tau_1'}{\partial q} - \frac{\tau_1'}{v_1} \left[1 + \frac{(1 - \tau_2)}{v_1} \frac{\partial v_1}{\partial \tau_2} \right] \frac{\partial \tau_2}{\partial q}. \tag{13}$$

Let $v_1 = A(\tau_2)\tau_1' + B(\tau_2)$. Then, from (10), we have

$$A(\tau_2) = \frac{\partial v_1}{\partial \tau_1'} = (1 - \tau_2)(T_s - \sigma) + \tau_1'(T_s - T_c) > 0. \tag{14}$$

Furthermore, $B(\tau_2) = v_1|_{\tau_1'=0} = (1 - \tau_2)T_s + \tau_2T_c > 0$. Hence, we have

$$\frac{\tau_1'}{v_1} \frac{\partial v_1}{\partial \tau_1'} = \frac{A(\tau_2)\tau_1'}{A(\tau_2)\tau_1' + B(\tau_2)} < 1. \tag{15}$$

In a similar manner, let $v_1 = C(\tau_1')\tau_2 + D(\tau_1')$. Then, for positive $C(\tau_1')$, by (15), Theorem 1 and Corollary 1, the right-hand side of (13) becomes positive. When $C(\tau_1') < 0$, we have

$$\left| \frac{(1 - \tau_2)}{v_1} \frac{\partial v_1}{\partial \tau_2} \right| = \left| \frac{C(\tau_1')(1 - \tau_2)}{C(\tau_1')\tau_2 + D(\tau_1')} \right| \leq \left| \frac{C(\tau_1')}{D(\tau_1')} \right| < 1,$$

because $D(\tau_1') > |C(\tau_1')|$. Consequently, we have $\partial S_1/\partial q > 0$ for all cases. □

In a similar manner, from (6), we have

$$S_2 = \frac{\tau_2(1 - \tau_1')}{v_2}.$$

Then, we have the following result for the relation between S_2 and q .

Theorem 3. *The saturation throughput of the well-behaved user, S_2 , is a decreasing function of the cheating rate q .*

Proof. By symmetry, from (13),

$$\frac{\partial S_2}{\partial q} = \frac{(1 - \tau'_1)}{v_2} \left[1 - \frac{\tau_2}{v_2} \frac{\partial v_2}{\partial \tau_2} \right] \frac{\partial \tau_2}{\partial q} - \frac{\tau_2}{v_2} \left[1 + \frac{(1 - \tau'_1)}{v_2} \frac{\partial v_2}{\partial \tau'_1} \right] \frac{\partial \tau'_1}{\partial q}. \quad (16)$$

Let $v_2 = A'(\tau'_1)\tau_2 + B'(\tau'_1)$. When $A'(\tau'_1)$ is positive, we have

$$\frac{\tau_2}{v_2} \frac{\partial v_2}{\partial \tau_2} = \frac{A'(\tau'_1)\tau_2}{A'(\tau'_1)\tau_2 + B'(\tau'_1)} < 1.$$

Hence, in all cases, the right-hand side of (16) is negative by virtue of Theorem 1 and Corollary 1. \square

5 Simulation Study

In this section, we perform a simulation study to verify our analysis. We use ns-2.34 with the MAC model in [12]. For saturation condition, we generate downlink UDP traffic of 6 Mb/s from the AP to each user. The default parameters used in our simulation is given in Table 1. For each given value of q , simulation run of 100 seconds has been performed for 20 times. Each point in figures is shown with a confidence level of 95%.

Table 1. Default parameters used in ns-2 simulations

802.11a modulation	BPSK (6Mbps)	Data rate	6 Mb/s
CW_{min}	15	CW_{max}	1023
RTS/CTS	Disabled	Thermal noise	-96 dBm
SINR threshold	10 dB	Rx threshold	-82 dBm

First, we consider the case when both of the users adopt the exponential backoff mechanism in Fig. 1. When the cheating rate q is zero, both users show the same throughput performance. However, as q increases, the throughput of the cheater increases while that of the normal user decreases, which agrees with our analysis. As q reaches one, the cheater takes most of the bandwidth share, and the throughput of the well-behaved user becomes almost zero.

Now, we take look into the case when both of the users use a fixed value of 15 for the contention window size in Fig. 2. Similarly as in Fig. 1, both users evenly share the bandwidth when q is zero. In addition, as q increases, the throughput of the cheater increases while that of the well-behaved user decreases in a similar manner. However, if we compare Fig. 2 with Fig. 1 carefully, it can be noticed that the rate of change in the throughput is smaller in Fig. 2. This difference results from the fact that the well-behaved user does not back off as q increases, but accesses the channel with a fixed contention window sizes (CW) of 15.

In Fig. 3, we consider the case when the cheater uses a fixed CW of 15 while the well-behaved user adopts the ordinary exponential backoff mechanism. As one can easily expect, the cheater accesses the channel in the most aggressive manner in this case

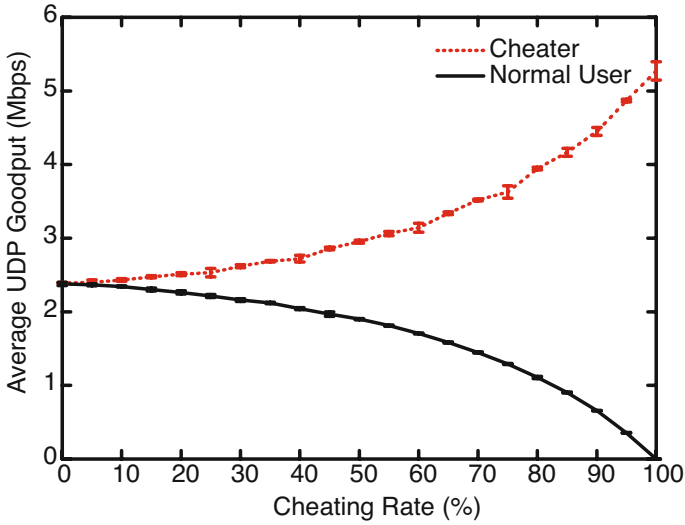


Fig. 1. Throughput performance vs. cheating rate when both users adopt the exponential backoff mechanism

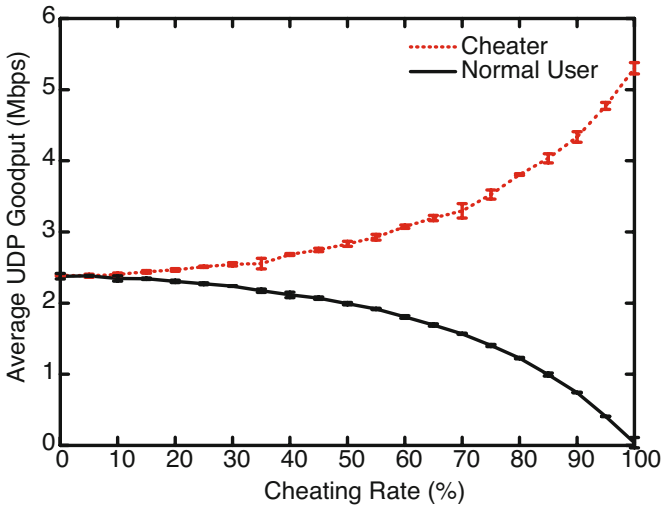


Fig. 2. Throughput performance vs. cheating rate when both users use a fixed CW of 15

among all three ones. Even when q is zero, there is a difference between the throughput of the cheater and that of the normal user. Since the cheater uses a fixed CW of 15, which is the minimum possible value for the normal user, the cheater can take more bandwidth than the well-behaved one in this case. As q increases, similarly as in the aforementioned cases, the cheater has more bandwidth share while the normal user loses its share.

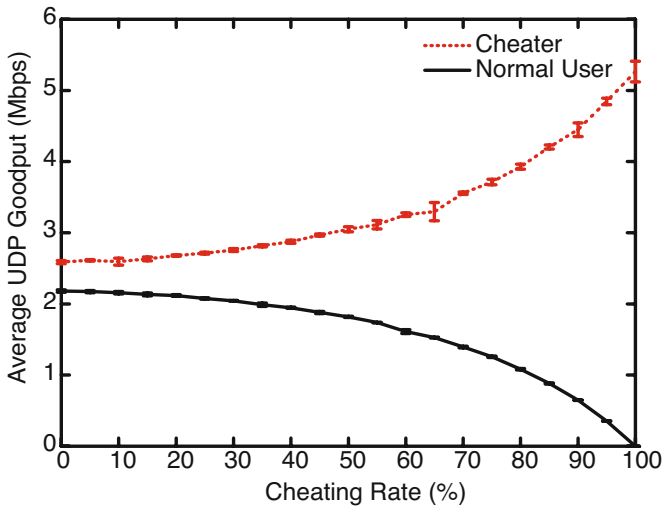


Fig. 3. Throughput performance vs. cheating rate when the cheater uses a fixed CW of 15 while the normal user adopts the exponential backoff mechanism

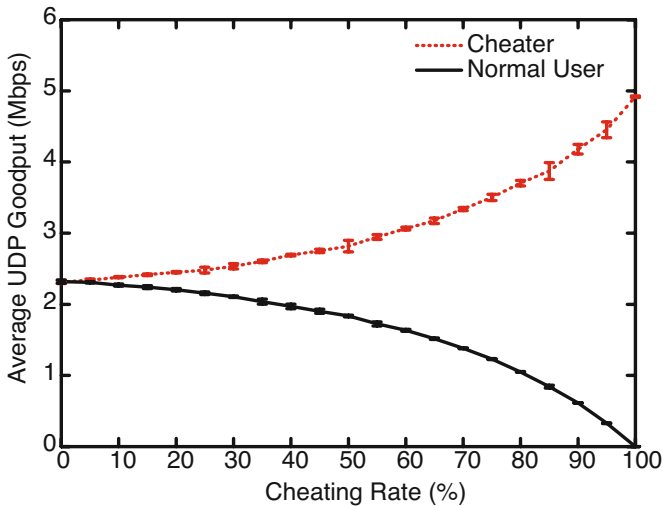


Fig. 4. Throughput performance vs. cheating rate when both users adopt the exponential backoff with RTS/CTS enabled

Finally, in Fig. 4 we consider the case when both users adopt the exponential backoff mechanism with Request to Send (RTS)/Clear to Send (CTS) enabled. Even though we have not considered RTS/CTS in our analysis, Fig. 4 shows that the trends are quite similar with those in the previous figures. Consequently, we can conclude that our

analysis is valid with the RTS/CTS procedure. A more detailed analysis of the network performance with the RTS/CTS mechanism will be an issue of future research.

6 Conclusion and Future Work

We have shown that a cheater can significantly increase its throughput by ignoring the carrier sense mechanism in IEEE 802.11 WLAN while a normal user will lose its throughput. In fact, our analysis shows that the carrier sense misbehavior corresponds to the case of virtually increasing the channel access probability of a cheater. Consequently, not only a malicious user, but also a selfish one are motivated to disregard the carrier sense procedure, which will result in significant degradation in throughput performance of well-behaved users. One important issue in future research is how to efficiently detect and penalize the carrier sense misbehavior of a selfish user to protect well-behaved ones from significant performance degradation.

References

1. Kyasanur, P., Vaidya, N.H.: Selfish MAC layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing* 4(5), 502–516 (2005)
2. Čagalj, M., Ganeriwal, S., Aad, I., Hubaux, J.P.: On selfish behavior in CSMA/CA networks. In: Proc. the 24th IEEE Conference on Computer Communications (INFOCOM 2005), Miami, FL, March 2005, pp. 2513–2524 (2005)
3. Konorski, J.: A game-theoretic study of CSMA/CA under a backoff attack. *IEEE/ACM Transactions on Networking* 14(6), 1167–1178 (2006)
4. Toledo, A., Wang, X.: Robust detection of selfish misbehavior in wireless networks. *IEEE Journal on Selected Areas in Communications* 25(6), 1124–1134 (2007)
5. Buttyán, L., Hubaux, J.P.: *Security and Cooperation in Wireless Networks*. Cambridge University Press, Cambridge (2007)
6. Pelechrinis, K., Yan, G., Eidenbenz, S., Krishnamurthy, S.: Detecting Selfish Exploitation of Carrier Sensing in 802.11 Networks. In: Proc. the 28th IEEE Conference on Computer Communications (INFOCOM 2009), Rio de Janeiro, Brazil (April 2009)
7. Fluhrer, S., Mantin, I., Shamir, A.: Weaknesses in the key scheduling algorithm of RC4. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 1–24. Springer, Heidelberg (2001)
8. Stubblefield, A., Ioannidis, J., Rubin, A.D.: A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Transactions on Information and System Security* 7(2), 319–332 (2004)
9. Bittau, A., Handley, M., Lackey, J.: The final nail in WEP's coffin. In: Proc. the 27th IEEE Symposium on Security and Privacy, Oakland, CA, May 2006, pp. 386–400 (2006)
10. Bellardo, J., Savage, S.: 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In: Proc. the 12th USENIX Security Symposium, Washington, DC, August 2003, pp. 15–27 (2003)
11. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 18(3), 535–547 (2000)
12. Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., Hartenstein, H.: Overhaul of IEEE 802.11 modeling and simulation in ns-2. In: Proc. the 10th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM 2007), Chania, Crete Island, Greece, October 2007, pp. 159–168 (2007)

Enhanced Bandwidth Allocation for TCP Flows in WiMAX Networks

Eun-Chan Park¹, Chunyu Hu², and Hwangnam Kim³

¹ Department of Information and Communication, Dongguk University, Korea

² Wireless Networking Business Unit, Broadcom Corporation, CA, USA

³ School of Electrical Engineering, Korea University, Korea

Abstract. In this paper, we propose a *bidirectional* bandwidth-allocation mechanism to improve TCP performance in the IEEE 802.16 WiMAX networks. According to the IEEE 802.16 standard, when serving a downlink TCP flow, the transmission of the uplink ACK, which is performed over a separate unidirectional connection, incurs additional bandwidth-request/allocation delay. Thus, it increases the round trip time of the downlink TCP flow and results in the decrease of throughput accordingly. First, we derive an analytical model to investigate the effect of the uplink bandwidth-request/allocation delay on the downlink TCP throughput. Second, we propose a simple, yet effective, bidirectional bandwidth-allocation mechanism that couples the bandwidth allocation for uplink and downlink connections by using either *proactive bandwidth allocation* or *piggyback bandwidth request*. The proposed scheme reduces unnecessary bandwidth-request delay and the relevant signaling overhead due to proactive allocation; meanwhile, it maintains high efficiency of uplink bandwidth usage by using piggyback request. Moreover, our proposed scheme is quite simple and practical; it can be simply implemented in the base station without requiring any modification in the subscriber stations or resorting to any cross-layer signaling mechanisms. The simulation results ascertain that the proposed approach significantly increases the downlink TCP throughput and the uplink bandwidth efficiency.

Keywords: IEEE 802.16e MAC, bandwidth request & allocation, TCP performance.

1 Introduction

The emerging broadband wireless access (BWA) network based on the IEEE 802.16e [1], called *Mobile WiMAX*, is one of the most promising solutions for the last mile broadband wireless access to support high data rate, high mobility, and wide coverage at low cost. The International Telecommunication Union (ITU) approved Mobile WiMAX as an International Mobile Telecommunication (IMT) advanced technology in October 2007. According to the WiMAX forum, the number of Mobile WiMAX users in the world is expected to grow up to 93 millions by 2012. On the other hand, TCP has been widely used in most

communication networks since the late 1980s, and it is still the most popular transport-layer protocol for reliable transmission in the Internet. Therefore, it is imperative to study and optimize the performance of TCP in Mobile WiMAX networks.

In this paper, we propose a solution for enhancing the TCP performance in Mobile WiMAX networks by means of efficient bandwidth allocation in the medium access control (MAC) layer. First, we show that the bandwidth-request delay, which is incurred in transmitting uplink TCP acknowledgements (ACKs), degrades the performance of the downlink TCP flow. Since the ACK packets are served with a separate uplink connection in Mobile WiMAX network, they require bandwidth-request/allocation procedure. This procedure incurs additional delay; therefore, the round trip time (RTT) of the downlink TCP flow is increased and the throughput is remarkably decreased. Moreover, we derive an analytical model for evaluating the effect of the bandwidth-request delay on the throughput of the downlink TCP flow. The numerical results based on the analysis model reveal that the downlink throughput decreases by about 20% ~ 30% under a typical configuration due to bandwidth-request delay.

In order to resolve this problem, we propose a framework of *bidirectional* connection that couples the bandwidth allocations for two unidirectional connections (one for downlink TCP data and the other for uplink TCP ACK). Within this framework, we propose a simple and effective bandwidth allocation mechanism that combines *proactive bandwidth allocation* with *piggyback bandwidth request*. The former allocates the bandwidth for the TCP ACK in a proactive manner; when a base station (BS) serves a downlink TCP data packet, the BS grants the bandwidth for the corresponding TCP ACK without any explicit request from the subscriber station (SS). The latter lets SS request bandwidth for the TCP ACK in a piggyback manner; SS carries the bandwidth-request for the subsequent ACKs in the header of on-going packet as long as there is ongoing uplink transmission.

The proposed approach decreases the bandwidth-request delay for the TCP ACK packets and reduces the overhead that is incurred in the bandwidth-request process. Implementing our proposed scheme is simple and practical, it is achieved by monitoring bandwidth-request queues managed by the BS without requiring any information or modification in the SS. This approach is a MAC-layer solution to improve the TCP performance; thus, it does not require any change in the TCP sender or receiver. Also, it can be incrementally deployed and widely extended to any centralized scheduling framework with a reliable transport protocol employing ACK mechanism. The OPNET [2] simulation results show that the proposed bidirectional approach increases the downlink TCP throughput up to about 40% compared with the conventional unidirectional bandwidth allocation, and it maintains high efficiency of the uplink bandwidth allocation.

There have been several proposals for efficient bandwidth request and allocation mechanisms in the IEEE 802.16 BWA networks in the literature [3,4,5,6]. They mostly focused on QoS scheduling algorithm and architecture, but they did not consider the TCP characteristics. TCP-aware uplink scheduling scheme

was recently proposed in [7], [8] to assure fair resource allocation among the competing uplink TCP flows. Also, the study in [9] dealt with the collision in the contention-based bandwidth request process, which may occur during the transmission of uplink TCP ACK. Our study differs from previous studies as follows: (i) we investigate the interaction between TCP and 802.16 MAC, and analyze the performance degradation in the downlink TCP flow resulting from the bandwidth allocation for the uplink TCP ACK, (ii) we propose the bidirectional bandwidth allocation aiming at increasing the throughput of the downlink TCP flow without decreasing the efficiency of the uplink bandwidth allocation, (iii) the proposed approach is transparent to a scheduling algorithm, i.e., any advanced downlink/uplink scheduling algorithm can be incorporated into the proposed framework to improve efficiency or QoS.

The rest of this paper is organized as follows. In Section 2 we briefly introduce the QoS scheduling framework of the IEEE 802.16, and we state the problem related to the bandwidth request and allocation for the TCP ACK. Next, we derive the analytical model of the TCP throughput by considering the bandwidth-request process in Section 3. In Section 4 we propose the framework and algorithm for the bidirectional bandwidth allocation. In Section 5 we evaluate the performance of the proposed approach via simulations. Finally, we conclude this paper in Section 6.

2 Problem Statement

2.1 IEEE 802.16 Scheduling Framework

This study considers the point-to-multipoint architecture of the IEEE 802.16 networks, where the communication between BS and SS is controlled by the BS. The transmissions are all made over unidirectional connections that are either downlink (DL) (from BS to SS) or uplink (UL) (from SS to BS). When a connection is established with the specific QoS requirements, the connection admission control comes into play at the BS based on the information of the advertised QoS requirements and the available resource. Once the connection is admitted, the BS schedules both DL and UL connections in a centralized way. The BS maintains two types of queues for scheduling, *data transmission queues* for DL connections and *bandwidth request queues* for UL connections. Based on the QoS requirements specified for each connection (e.g., the tolerable delay and the minimum reserved rate), the BS schedules the DL connections with the transmission queues and UL connections with the request queues, independently. Unlike the DL connections, the bandwidth for the UL connections is allocated on a reservation-basis or on a request-basis depending on the scheduling class. After completing the scheduling process, the BS generates and broadcasts DL/UL MAP messages that contain two dimensional (time and frequency) resource allocation information. When receiving the DL/UL MAP, SS decodes a DL frame and transmits a UL frame in the specified time and frequency of the OFDMA/TDD (Orthogonal Frequency Division Multiple Access with Time Division Duplex) frame.

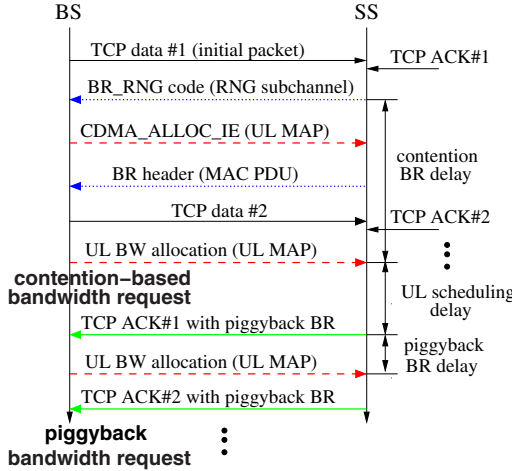


Fig. 1. Bandwidth-request procedure for TCP ACK packets; contention-based request and piggyback request

2.2 Bandwidth Request for TCP ACK

Considering TCP ACK packets are served with a best-effort (BE) connection, there are two standardized bandwidth-request mechanisms for serving them [1]: the *contention-based request* and the *piggyback request*, which are referred to as *contention* and *piggyback* hereafter, respectively.

In the *contention* method, the SS takes the following four-step request-response procedure for transmitting a TCP ACK, as illustrated in Fig. 1; (i) the SS picks a random bandwidth-request ranging (BR_RNG) code, which is modulated into a dedicated contention-free ranging channel and it is delivered to the BS; (ii) the BS detects the BR_RNG code, and then sends a CDMA_Allocation_IE message in the UL MAP to inform SS of the transmission region for a BR message; (iii) the SS sends a stand-alone BR MAC header as a MAC protocol data unit (MPDU) that specifies the required amount of bandwidth; (iv) the BS allocates the required bandwidth by sending the UL MAP back to the SS. Finally, the SS can transmit its TCP ACK packet by using the allocated bandwidth. This procedure inevitably incurs a processing delay that is approximately two tenths of a millisecond. Sometimes the delay may increase up to a few hundred milliseconds due to the collisions and the subsequent backoff/retransmissions (when two or more SSs choose the same BR_RNG code and simultaneously send them). In this case, the delay can cause TCP-level time-out and retransmission, which drastically decrease the TCP throughput.

On the other hand, the SS can deliver the ACK packets via the *piggyback* method. For the MAC frames backlogged in the transmission queue of the SS, the corresponding BR message can be piggybacked in the sub-header of the

¹ For simplicity, we do not consider the delayed ACK mechanism [10] or the fragmentation/packing of MAC service data unit (MSDU) in Fig. 1.

on-going MAC frame. Fig. 1 shows that the second TCP ACK is delivered by the *piggyback*, while the first ACK is delivered by the *contention*. Compared to the *contention* method, the *piggyback* method neither requires contention for the BR opportunity, nor incurs long delay. Moreover, the *piggyback* method reduces signaling overhead; specifically, the size of the BR MAC header for the *contention* is 6 bytes, while the size of the sub-header for the *piggyback* is 2 bytes [1]. Consequently, it is more desirable to use the *piggyback* method for delivering TCP ACKs.

However, the *piggyback* method is only available when there exists at least one backlogged MAC frame in the transmission queue at the instant of generating a new MAC frame that contains TCP ACK. The *piggyback* method is not always available due to the following reasons:

- TCP data packets are generated and delivered in a bursty fashion, so the corresponding ACK packets are not regularly or periodically generated, i.e., the transmission queue in the SS is occasionally empty.
- When packet loss or TCP time-out occurs (which frequently happens in wireless networks), there is no choice but to perform the *contention* method to serve ACK packets because there is no on-going UL frame.
- The first ACK packet of the first data packet within a certain congestion window may not use the *piggyback*.

3 Modeling TCP Throughput with Bandwidth-Request Process

3.1 Model Derivation

We derive the TCP throughput model by considering and analyzing the effect of the BR delay on the throughput. We consider that a TCP connection is established to download an L -byte object. Here, we make several reasonable assumptions; (i) the wireless link between the BS and the SS is a bottleneck link and its capacity is constant, (ii) the buffer in the BS is properly provisioned to prevent buffer overflow, (iii) a retransmission mechanism, hybrid automatic repeat request (HARQ), recovers the wireless channel error and it assures in-order delivery of the MPDUs according to the IEEE 802.16 specification [1]. Let us denote p_e as the final target packet error rate with the HARQ retransmissions and denote W_{th} and $W_{max} (> W_{th})$ as *slow start threshold* and *advertised window size*, respectively. Then, we model the TCP congestion window $w(k)$ at the k th RTT stage as the Markov chain and we represent the state transition probabilities as:

$$\begin{aligned}
 \text{Prob}[w(k+1) = 2^{1/b}w \mid w(k) = w < W_{th}] &= (1 - p_e)^w \\
 \text{Prob}[w(k+1) = w + 1/b \mid w(k) = w \geq W_{th}] &= (1 - p_e)^w \\
 \text{Prob}[w(k+1) = W_{max} \mid w(k) = W_{max}] &= (1 - p_e)^w \\
 \text{Prob}[w(k+1) = w/2 \mid w(k) = w] &= 1 - (1 - p_e)^w
 \end{aligned} \tag{1}$$

Here, b denotes the number of packets acknowledged by one received ACK packet (if the delayed ACK mechanism is used, then the TCP receiver sends one cumulative ACK for two consecutively received packets, i.e., b is 2). This stochastic process models three phases of the TCP congestion window: slow-start, congestion avoidance, and fast recovery, but it neglects the TCP time-out due to the assumptions of a large buffer size and the HARQ retransmissions. Next, we calculate the number of RTT stages required for downloading the L -byte object, defined as N . We consider that packet losses are not correlated among the back-to-back transmissions within a congestion window (because the buffer size is large enough to avoid buffer overflow and the packet error in the wireless channel is random and irrelevant to the queuing discipline). We also consider that a lost packet, even after the HARQ retransmissions, is recovered by a single TCP retransmission with probability close to one since $p_e \ll 1$. Thus, we can represent N as

$$N = \operatorname{argmin}_n \sum_{k=1}^n w(k) > L' + \sum_{k=0}^{L'} k \binom{L'}{k} p_e^k (1 - p_e)^{L'-k}, \tag{2}$$

where $L' = \lceil L/MSS \rceil$ and MSS (byte) denotes the maximum segment size of TCP. The first and second terms on the right side of (2) represent the initial transmission and the retransmission of the TCP packets. The second term on the right side of (2) is equal to the mean of the binomial distribution; so the right side of (2) becomes $L'(1 + p_e)$.

Next, we model the RTT at the k th stage, $RTT(k)$. Fig. 2 depicts the TCP timing diagram between the sender (server) and the receiver (SS). Note that the BS is located between the server and the SS but it is not explicitly shown in Fig. 2. We define R (byte/sec) as the effective capacity to process the TCP payload. At the receiver-side, $RTT(k)$ can be considered as the difference between the time when the receiver starts receiving the first packet in $w(k)$ and the time when it does in $w(k + 1)$, i.e.,

$$RTT(k) = \frac{w(k)MSS}{R} + t_{idle}(k), \tag{3}$$

where $t_{idle}(k)$ is the idle time between the time when finishing receiving the last packet in the k th RTT stage and the time when starting receiving the first packet in the $(k + 1)$ th RTT stage. On the other hand, the RTT can be modeled at the sender-side as the difference between the time when the sender starts transmitting the first packet in the congestion window and the time when the sender receives the corresponding ACK packet, i.e.,

$$RTT(k) = \frac{MSS}{R} + t_{br}(k) + t_q(k) + RTT_{min}. \tag{4}$$

Here, $t_{br}(k)$ denotes the BR delay of the ACK for the first data packet in the k th RTT stage, and it can be modeled as

$$t_{br}(k) = \begin{cases} T_{ct} & \text{if } t_{idle}(k - 1) > 0, \\ T_{pb} & \text{if } t_{idle}(k - 1) = 0, \end{cases} \tag{5}$$

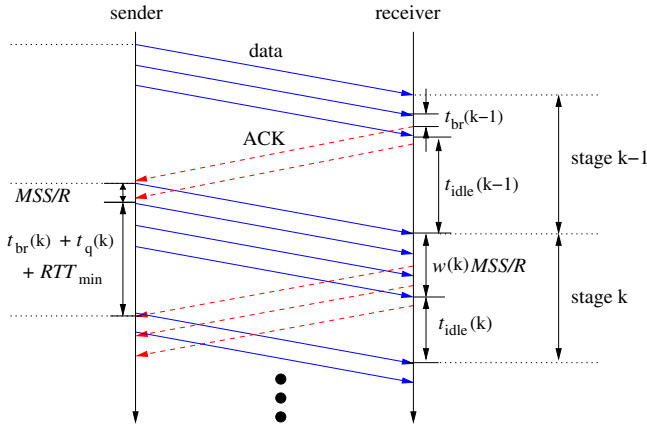


Fig. 2. TCP timing diagram at sender-side and receiver-side

where T_{ct} and T_{pb} are the contention-based and the piggyback-based BR delays, respectively. We define $t_q(k)$ as the queuing delay in the BS for the first data packet in the k th RTT stage and we model it as the accumulated backlog until the $(k - 1)$ th RTT stage divided by the service rate at the k th stage, i.e.,

$$t_q(k) = \sum_{n=1}^{k-1} \left[w(n) \frac{MSS}{R} - RTT(n) \right]^+, \tag{6}$$

where $[x]^+ = \max(0, x)$. In (4), RTT_{min} accounts for several components of the RTT except t_{br} and t_q , e.g., the propagation delay over wired links between the sender and the BS, the DL/UL scheduling delay in BS/SS, the ACK transmission delay, and several other processing delays. Although RTT_{min} may vary, we assume that it is constant in order to focus on the throughput reduction due to t_{br} .

The RTT is matched at both the sender-side and the receiver-side, thus, $t_{idle}(k)$ can be represented from (3) and (4) as:

$$t_{idle}(k) = \left[RTT_{min} + t_q(k) + t_{br}(k) - (w(k) - 1) \frac{MSS}{R} \right]^+. \tag{7}$$

Once $w(k)$ is obtained from the stochastic process described in (1); $t_q(k)$ and $t_{idle}(k)$ can be obtained from (6) and (7), respectively, and $RTT(k)$ is determined by (4). Finally, the average throughput, \overline{TH} , is given as

$$\overline{TH} = \frac{\sum_{k=1}^N w(k)MSS}{\sum_{k=1}^N RTT(k)}, \tag{8}$$

where N is given in (2). Note that the average throughput \overline{TH} is not represented in a closed form, but it can be numerically obtained.

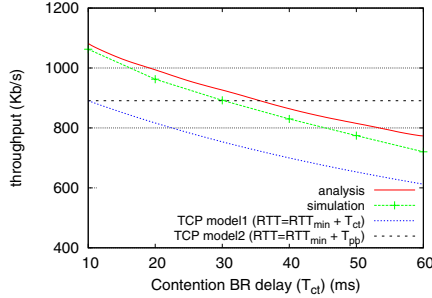


Fig. 3. Effect of bandwidth-request delay on TCP throughput

3.2 Model Validation

We validate the derived model by comparing its results with the simulation results. We also compare the analysis results with the well-known TCP throughput model [11], which is characterized by the RTT and packet loss rate as;

$$\overline{TH} = \min \left(\frac{W_{max}MSS}{RTT}, \sqrt{\frac{3}{2b}} \frac{MSS}{RTT\sqrt{p_e}} \right). \tag{9}$$

Here, we consider b to be one without considering the delayed ACK mechanism for simplicity.

First, we observe the effect of the BR delay on the TCP throughput. We consider a typical configuration such that $W_{max} = 64$ KB, $MSS = 1$ KB, $L = 1$ MB, $RTT_{min} = 100$ ms, $T_{pb} = 10$ ms, $R = 5$ Mb/s, and $p_e = 0.01$. Figure 3 compares the analysis results with two TCP models, TCP model1 and TCP model2, as well as with the simulation results to validate the analysis model. The results of TCP model1 and TCP model2 are obtained from (9) by setting their RTTs such that $RTT = RTT_{min} + T_{ct}$ (TCP model1) and $RTT = RTT_{min} + T_{pb}$ (TCP model2),² respectively. They are intended to represent the cases where the bandwidth is requested by either only the contention or the piggyback. Compared to the case of $T_{ct} = 10$ ms, the throughput obtained from the simulation and the analysis results is approximately decreased by 32% and 29% when T_{ct} is increased to 60 ms, respectively. The analytical throughput is slightly higher than that of the simulation result, which results from the fact that the model does not consider the delay variation due to the HARQ and the burst behavior of TCP packet loss. However, the throughputs with TCP model1 and TCP model2 deviate from the simulation results remarkably. The main reason of this deviation is the assumption made in deriving (9) i.e., once a packet is lost, the subsequent packets are dropped due to the buffer-overflow until the end of the given RTT stage. This assumption is reasonable in wired networks where the routers' buffers are

² In this configuration, we observed from the simulation that the queuing delay is negligible compared to RTT_{min} so it is not included in RTT calculation.

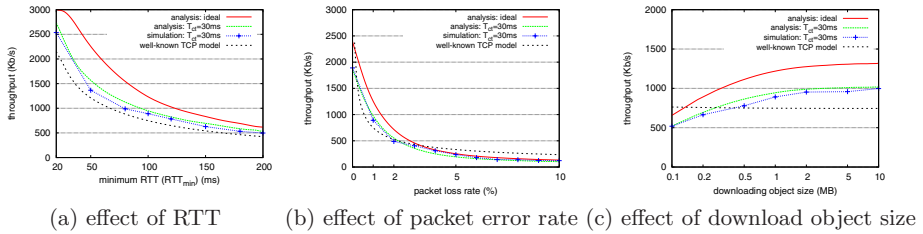


Fig. 4. Validation of TCP throughput model with various parameters

managed by the drop-tail queuing discipline; however, it is no longer valid in the wireless networks where packet loss is not highly related to the buffer-overflow, but it occurs randomly. On the other hand, TCP model2 gives constant throughput, regardless of the change of the contention BR delay, T_{ct} . If $T_{ct} > 30$ ms, then TCP model2 overestimates the throughput compared to the simulation results. However, it underestimates the throughput when $T_{ct} < 30$ ms. These results in Fig. 3 confirm that the existing TCP throughput models cannot capture the effect of the contention-based BR delay but our analysis model is effective to do that.

Our derived analysis model can also evaluate the effect of various system parameters such as RTT (RTT_{min}), packet error rate (p_e), and download object size (L). Fig. 4 shows these effects on the achievable TCP throughput. Here, we set T_{ct} to the typical value of 30 ms, and we compare the analysis results with the simulation results. Additionally, Fig. 4 shows the maximum theoretical throughput that can be obtained from our analysis model by setting $T_{ct} = 0$ and $p_e = 0$. We also compare these results with that of the TCP model [11], where RTT in (9) is set as the average value from the simulation results. We set the default values of the parameters as $RTT_{min} = 100$ ms, $p_e = 1\%$, $L = 1$ MB, and $R = 5$ Mb/s. From Fig. 4 we observe the following:

- The analysis results agree well with the simulation results for the wide range of various system parameters, which confirms that the analysis model is effective and accurate.
- The existing TCP model underestimates the throughput in most cases. Furthermore, it fails to represent the effect of L , i.e., it gives a constant throughput regardless of the value of L , because the existing TCP model is intended to get the steady-state throughput without considering the slow-start phase of the TCP.
- Compared to the ideal case without the BR delay, the actual TCP throughput is decreased by about 10% ~ 23% for the whole configuration. Also the relative throughput, defined as the actual throughput divided by the ideal throughput, is decreased as RTT_{min} or L decreases. This implies that the BR delay effect is amplified when the RTT or the object size is small.

4 Bidirectional Bandwidth Allocation

We propose the bidirectional bandwidth allocation for the DL TCP data and the UL TCP ACK to reduce both the BR delay and the overhead. First, we propose a preliminary solution that proactively allocates the bandwidth for the UL TCP ACK when the BS serves the DL TCP data packet. Next, we elaborate on how this mechanism improves the efficiency of the UL bandwidth allocation by combining the proactive allocation with the piggyback request.

4.1 Proactive Bandwidth Allocation

The starting point of bidirectional bandwidth allocation is that a TCP flow essentially involves the bidirectional packet transmission, i.e., the sender transmits the data packets to the receiver while the receiver transmits the ACK packets to the sender. Also, the transmission of the TCP ACK is related to the transmission of the TCP data; no TCP ACK packet is generated until the TCP data packet is delivered to the receiver. However, the process of bandwidth allocation standardized in IEEE 802.16 works in a unidirectional way; the UL bandwidth allocation is completely independent of the DL bandwidth allocation. Under this rationale, we propose a bidirectional connection, where the bandwidth allocation for the UL TCP ACK is associated with the transmission of the DL TCP data. We consider the *proactive bandwidth allocation* (we call it *proaction*) mechanism as a naive approach. The BS scheduler proactively allocates bandwidth for the corresponding UL ACK packet whenever the DL TCP data packet is served by the BS. Thus, it is not necessary for the SS to request bandwidth and the *proaction* can remove the BR delay and the overhead that are caused by transmitting the UL ACK packets. Consequently, the DL TCP throughput can be increased.

However, this approach has two major drawbacks. First, if the delayed ACK [10] mechanism is used it wastes the UL bandwidth. The delayed ACK mechanism, which is implemented in most TCP protocols, lets the TCP receiver not send the ACK packets immediately after receiving the TCP data packets but the receiver waits for the arrival of the next in-order TCP data packet up to 500 ms. Combined with the cumulative ACK, the delayed ACK mechanism can effectively reduce the amount of ACK traffic. Roughly speaking, the receiver sends every other ACK packet on receiving TCP data packets. Consequently, the *proaction* may unnecessarily allocate bandwidth and significantly decreases the UL bandwidth efficiency. Second, the *proaction* cannot determine the accurate amount of bandwidth-request if the MSDU (TCP data packet or ACK packet) is fragmented or packed. In the IEEE 802.16 MAC, the MSDU fragmentation/packing frequently occurs in the scheduling and automatic repeat request (ARQ) operation. If the BS serves the fragmented or packed TCP data packets, then it can hardly predict the amount of bandwidth required for the SS to serve the corresponding ACK packets. For these reasons, the *proaction* mechanism cannot be considered as a practical solution for the bidirectional bandwidth allocation.

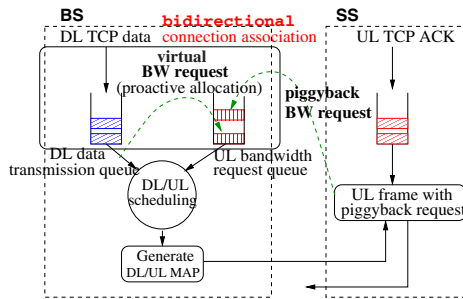


Fig. 5. Schematic diagram of bidirectional bandwidth allocation combining proactive bandwidth allocation with piggyback bandwidth request

4.2 Hybrid Approach

To overcome the drawbacks of the proactive bandwidth allocation, we propose a hybrid approach that combines the proactive bandwidth allocation with the piggyback bandwidth request. The proposed approach basically employs the *piggyback* method, and utilizes the *proaction* method only if the *piggyback* is not available. The *piggyback* does not have any problem involved to the delayed ACK mechanism and the fragmentation/packing since it allocates the bandwidth reactively (the SS first determines the amount of required bandwidth and then it requests the bandwidth). The basic idea is to first limit the usage of the *proaction* for efficient usage of the UL bandwidth, and then to replace the *contention* with the *proaction* for enhancing the DL throughput. The key point is determining the type of UL bandwidth allocation (*proaction* or *piggyback*) and determining the amount of bandwidth request.

Fig. 5 presents how the proposed approach works. Note the UL bandwidth request queue in the BS is managed for each connection. When serving a DL frame, the BS scheduler checks the associated UL request queue. If the UL request queue is empty, the scheduler puts a new bandwidth request to the request queue on behalf of the SS, i.e., the scheduler performs the *proaction*. The amount of bandwidth-request is set to be equal to the amount required to send one TCP ACK including the MAC header. On the other hand, when transmitting an UL frame, the SS checks whether the backlogged traffic is still present in its transmission queue. If the transmission queue is not empty, then the SS requests bandwidth in a piggyback manner at the amount of the backlogged traffic. In this way, the proposed hybrid approach increases the UL bandwidth efficiency due to the *piggyback* while decreasing BR delay due to the *proaction*.

The main strength of the proposed approach is that it does not require any modification of the SS and it can be simply implemented in the BS. Moreover, the proposed mechanism does not have any control parameters on which its performance depends, and thus it is unnecessary to tune the parameters. The proposed mechanism only monitors the state of the UL request queue maintained in the BS to determine the type of bandwidth-request, so it neither requires information about the transmission queue of the SS nor results in additional

signaling overhead between the BS and the SS. In addition to simplicity, this approach can be incrementally deployed because it does not require any changes of the SS. The proposed approach can also be incorporated with any scheduling algorithm to further improve overall utilization, fairness, or QoS.

5 Simulation

5.1 Simulation Setup

In the simulations, we consider the OFDMA/TDD PHY where the frame duration is 5 msec, the number of DL/UL symbols are 29/18, and the base frequency and the channel bandwidth are 2.5 GHz and 10 MHz, respectively. The wireless channel is modeled by using the empirical COST-231 HATA model [12], log-normal shadowing, and the ITU channel model [13] to consider the multipath fading effect. We implement the adaptive modulation and coding scheme in the simulator. Depending on the signal to interference noise ratio (SINR), the modulation and coding rate are dynamically changed among the followings: QPSK (1/12, 1/8, 1/4, 1/2, 3/4), 16QAM (1/2), 64QAM (2/3, 3/4, 5/6) for the downlink and QPSK (1/12, 1/8, 1/4, 1/2, 3/4), 16QAM (1/2, 3/4) for the uplink. We emulate the HARQ and the ARQ mechanisms such that the target packet error rate of the HARQ is 1 %, the maximum number of the HARQ retransmissions (excluding the initial transmission) is three, and the retransmission delays of the HARQ and the ARQ are 30 ms and 100 ms, respectively. Also, they are modeled so that MSDUs are delivered in-order for reducing TCP retransmissions. The contention-based BR delay is uniformly set between 25 ms to 50 ms; the collision probability of BR_RNG code and its timer value are set to 1 % and 100 ms, respectively. We set the maximum TCP segment size to 1500 bytes and we use the delayed ACK mechanism. The minimum RTT, RTT_{min} is set to 100 ms. These configurations are typical operation scenarios for the TCP flows over the mobile WiMAX networks, as recommended by the IEEE 802.16m task group [14]. Note that the simulation results are averaged over ten instances of the simulation with different random seeds.

In the simulation study, we consider the following three bandwidth-request/allocation algorithms and we compare their performance;

- (i) **UNI**: a conventional unidirectional approach as described in Sec. 2.2,
- (ii) **blind-BI**: a bidirectional approach using the *proaction* without the *piggyback*,
- (iii) **adapt-BI**: the proposed bidirectional approach that adaptively switches between the *proaction* and the *piggyback* depending on the state of the BR queue.

Together with simulation results of these algorithms, we will present the theoretical maximum throughput derived from the analysis model in Section 3, it is named as **ideal**. The performance metrics are set as:

- **UL MAC-to-MAC delay**: the time interval between generating MSDU on the SS and receiving it on the BS.

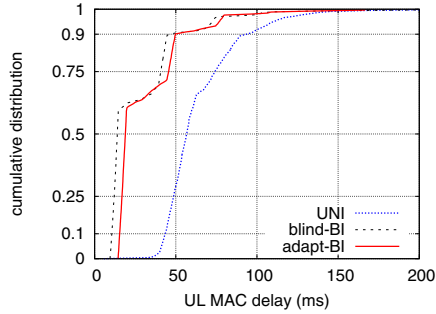


Fig. 6. Uplink MAC-to-MAC delay of UNI, blind-BI, and adapt-BI

- UL bandwidth allocation efficiency: B_{used}/B_{alloc} , where B_{used} is the cumulative amount of UL bandwidth that is actually used to transmit the TCP ACKs and B_{alloc} denotes the cumulative amount of bandwidth allocation (including the MAC header and the bandwidth-request message).
- DL throughput: the average goodput, which is calculated as the download object size divided by the download completion time.

5.2 Tradeoff between Performance and Efficiency

In the first simulation, we focus on the tradeoff between performance and efficiency in the bandwidth allocation process. Here, we consider FTP download and we set the download object size to 1 MB.

First, we observe the UL MAC-to-MAC delay of UNI, blind-BI, and adapt-BI, whose cumulative distributions are shown in Fig. 6. The delay of blind-BI is minimized due to proactive bandwidth allocation, but that of UNI is much larger than the others due to the contention-based request. However, the adapt-BI considerably decreases the delay, which is at least 2 times smaller than that of UNI and is slightly higher than that of blind-BI. Specifically, the median delays of UNI, blind-BI, and adapt-BI are 56.6, 14.0, and 18.9 ms, and the 90th-percentile delays are 94.1, 46.7, and 49.8 ms, respectively. The sudden increase of the delay in Fig. 6 is caused by the HARQ/ARQ retransmissions.

Next, we evaluate the efficiency of the UL bandwidth allocation. Table 1 lists B_{alloc} , B_{used} , as well UL efficiency and DL throughput. The UNI utilizes the UL bandwidth efficiently, i.e., the difference between B_{alloc} and B_{used} is slight. However, in the case of the blind-BI, B_{alloc} is almost two times higher than B_{used} , i.e., almost half of the allocated bandwidth is wasted. The bandwidth wastage of blind-BI is remarkably higher than the other algorithms because it proactively allocates the bandwidth, regardless of whether or not the SS has packets to send. As indicated in Table 1, the bandwidth allocation waste is minimized by the adapt-BI; it is smaller than that of the UNI because the UNI has to send a 6-byte bandwidth-request message in the *contention* phase, which is unnecessary in the adapt-BI.

Table 1. Uplink efficiency and downlink throughput of UNI, blind-BI, and adapt-BI

algorithm	B_{alloc} (Kbyte)	B_{used} (Kbyte)	UL efficiency	DL throughput (Mb/s)
UNI	18.6	16.4	0.88	0.77
blind-BI	32.7	16.1	0.49	1.06
adapt-BI	17.9	16.3	0.91	1.01

In summary, the results in Fig. 6 and Table 1 confirm the following:

- Compared to the UNI, the adapt-BI decreases the average UL MAC-to-MAC delay by more than two times, and so it increases the average DL throughput by about 31%.
- the blind-BI achieves small gain of DL throughput over the adapt-BI at the significant cost of efficiency of the UL bandwidth allocation; almost half of the UL bandwidth is wasted to increase the DL throughput by about 5% compared to adapt-BI.
- Unlike the blind-BI, the adapt-BI maintains high efficiency of the UL bandwidth allocation (higher than blind-BI by 42%); meanwhile, its DL throughput is comparable with that of the blind-BI.

5.3 Effect of RTT

In this simulation, we study the effect of RTT, which is a key factor affecting the TCP throughput. For this purpose, we change RTT_{min} , which can be set to an arbitrary value by excluding the variable components of RTT such as BR delay and scheduling/queuing delay, from 20 ms to 200 ms. Here, L is set to 1 MB.

As shown in Table 2, the UL MAC-to-MAC delay and the UL efficiency are almost insensitive to the change of RTT_{min} . Regardless of RTT_{min} , the adapt-BI considerably reduces the UL delay compared to the UNI, while it remarkably increases the UL efficiency compared to the blind-BI. As RTT_{min} increases, its effect on the throughput surpasses that of the BR delay. Therefore, the effect of the BR delay on the throughput is alleviated, i.e., the throughput increase of adapt-BI/blind-BI over UNI decreases as RTT_{min} increases. For example, if $RTT_{min} = 20$ ms, the adapt-BI gives higher throughput than the UNI by about 39%, but the throughput gain is decreased by 18% if $RTT_{min} = 200$ ms. These simulation results in Table 2 confirm the outstanding performance of adapt-BI in terms of bandwidth request delay, uplink bandwidth efficiency, and downlink throughput.

Next, we compare the ideal throughput calculated from the analysis model with those obtained from simulations. Table 2 shows that the throughput of adapt-BI is close to the ideal throughput; the slight difference between them is mainly caused by the fact that the analysis model does not include the HARQ/ARQ processing delays. Also, we can check the validity of the analysis model by comparing its ideal throughput with the throughput of blind-BI. Apart from the UL bandwidth efficiency, the blind-BI is considered to be the

Table 2. Performance comparison of UNI, blind-BI, and adapt-BI with various values of RTT_{min}

RTT_{min} (ms)	UL MAC delay (ms)			UL efficiency (%)			DL throughput (Mb/s)			
	UNI	blind -BI	adapt -BI	UNI	blind -BI	adapt -BI	UNI	blind -BI	adapt -BI	ideal
20	62.0	26.8	31.1	87.6	48.9	91.0	1.19	1.76	1.65	1.79
50	64.9	26.9	30.9	87.9	49.1	90.9	0.98	1.42	1.35	1.47
100	65.5	27.0	31.3	88.1	49.7	91.0	0.76	1.04	1.01	1.11
150	65.8	26.7	30.7	88.3	50.2	91.1	0.60	0.78	0.74	0.81
200	65.5	26.6	31.6	88.2	51.2	91.2	0.51	0.63	0.60	0.67

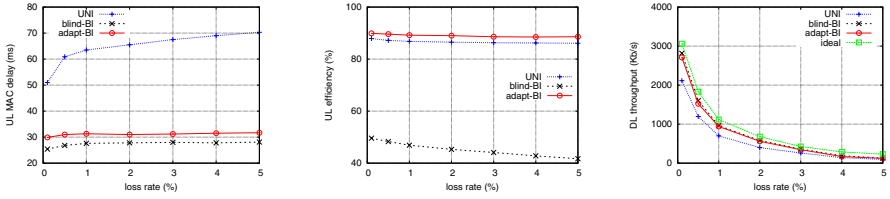
best solution that can maximize the DL throughput by minimizing the BR delay. The result in Table 2 that the analytical throughput is nearly equal to the throughput of blind-BI reconfirms the validity of our analysis model.

5.4 Effect of Packet Loss Rate

In this simulation, we evaluate the performance of the proposed mechanism with various values of packet loss rate. To focus on the effect of packet loss on the TCP throughput, we disable the retransmission mechanisms in PHY/MAC layer (HARQ and ARQ) and we randomly drop packets with probability of p ranging from 0.1% to 5%³. Here, we set L and RTT_{min} to 1 MB and 100 ms, respectively.

Fig. 7 compares several performance indices of the three algorithms with various values of p . It is noteworthy from Fig. 7(a) that the UL delay of UNI increases as p increases, but those of blind-BI and adapt-BI change very little with respect to p . The reason is as follows. A packet loss can result in the TCP time-out or increase the burstiness of packet transmission, then the packet loss possibly makes the transmission queue of the SS empty. Accordingly, the probability of the contention-based BR is increased and the UL MAC delay of UNI is also increased. On the other hand, the delays of blind-BI and adapt-BI are nearly irrespective of packet loss because of the proactive bandwidth allocation, i.e., even when the transmission queue of the SS is empty, the bandwidth is allocated without a contention-based request. Next, we can observe the UL efficiency from Fig. 7(b). As p is increased from 0.1% to 5%, the efficiencies of UNI and adapt-BI are both decreased by 1.8%, however, that of blind-BI is decreased by 7.9%. The blind-BI unconditionally allocates bandwidth for the UL TCP ACK, regardless of whether or not the DL TCP data packet is successfully delivered to the SS, so the allocated UL bandwidth may be wasted if the DL packet loss occurs. Consequently, the delay of blind-BI increases in proportion to the packet loss rate. Next, we investigate the effect of p on the throughput from Fig. 7(c). As was expected, the throughputs of all the three algorithms decrease rapidly as p increases. Although the absolute throughput gain of adapt-BI over UNI, which is the throughput difference between them, decreases with respect

³ If HARQ and ARQ are enabled, most of the packet losses due to the wireless channel errors are recovered, so we cannot arbitrarily set the packet loss rate.



(a) uplink MAC-to-MAC delay (b) uplink bandwidth efficiency (c) downlink throughput

Fig. 7. Performance comparison of UNI, blind-BI, and adapt-BI with various values of packet loss rate

to the increase of p , and the relative throughput gain, which is the throughput of adapt-BI divided by that of UNI, remains within 23% ~ 40% for the entire range of p . We also observe from Fig. 7(c) that the throughput of blind-BI/adapt-BI is not quite deviated from the ideal throughput.

6 Conclusion

In this paper, we have proven that the DL TCP performance is degraded in the IEEE 802.16 wireless networks due to the bandwidth-request delay for transmitting the UL ACK. Moreover, we have derived the analytical model through which we can quantitatively analyze the effect of the bandwidth-request delay on the TCP throughput. The model is useful for predicting the maximum throughput gain that can be achieved by an ideal bandwidth request/allocation mechanism. To remove the unnecessary bandwidth-request delay and overhead that are involved in transmitting the UL ACK, we have proposed the bidirectional bandwidth allocation framework and the hybrid approach that combines the proactive bandwidth allocation with the piggyback bandwidth request schemes. Due to proactive bandwidth allocation, the proposed approach can reduce both bandwidth-request delay and overhead, and it can increase the DL throughput. At the same time, it can increase the efficiency of the UL bandwidth allocation due to the piggyback request, which is performed in a reactive manner. The simulation results have indicated that the proposed hybrid approach significantly increases the DL TCP throughput (by up to 40% compared to the case without the proactive allocation) as well as the UL bandwidth efficiency (by about two times compared to the case without the piggyback request). The advantages of the proposed scheme are that it is very simple and practical, and it requires neither changes of the SS nor additional signaling mechanism between the BS and the SS.

Acknowledgment

This work was supported in part by the IT R&D program of MKE/IITA [2008-F015-02, Research on Ubiquitous Mobility Management Methods for Higher Service Availability].

References

1. IEEE 802.16 WG, IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems, Amendment 2, IEEE 802.16 Standard (December 2005)
2. OPNET WiMAX Model Development Consortium, "OPNET network simulator with WiMAX model (2007), <http://www.opnet.com/WiMax>
3. Wongthavarawat, K., Ganz, A.: Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. *International Journal of Communication systems* 16, 81–96 (2003)
4. Cicconetti, C., Lenzini, L., Mingozzi, E., Eklund, C.: Quality of service support in IEEE 802.16 networks. *IEEE Network* 20, 50–55 (2006)
5. Liu, Q., Wang, X., Giannakis, G.B.: A cross-layer scheduling algorithm with QoS support in wireless networks. *IEEE Trans. on Vehicular Technology* 55, 839–847 (2006)
6. Park, E.-C., Kim, H., Kim, J.-Y., Kim, H.-S.: Dynamic bandwidth request-allocation algorithm for real-time service in IEEE 802.16 broadband wireless access networks. In: *Proceedings of IEEE INFOCOM* (2008)
7. Kim, S., Yeom, I.: TCP-aware uplink scheduling for IEEE 802.16. *IEEE Communications Letters* 11, 146–148 (2006)
8. Rath, H.K., Karandikar, A., Sharma, V.: Adaptive modulation-based tcp-aware uplink scheduling in IEEE 802.16 networks. In: *Proceedings of IEEE ICC* (2008)
9. Kim, E., Kim, J., Kim, K.S.: An efficient resource allocation for TCP service in IEEE 802.16 wireless MANs. In: *Proceedings of IEEE Vehicular Technology Conference (VTC)-Fall*, pp. 1513–1517 (2007)
10. Braden, R.: Requirements for Internet hosts – Communication layers. *IETF RFC* 1122 (October 1989)
11. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP throughput: A simple model and empirical validation. In: *Proceedings of ACM SIGCOMM*, pp. 303–314 (1998)
12. Blaunstein, N.: *Radio Propagation in Cellular Networks*. Artech House (1999)
13. ITU-R Task Group 8/1, Guidelines for evaluation of radio transmission technologies for IMT-2000, Recommendation ITU-R M.1225 (1999)
14. IEEE 802.16 WG, Draft IEEE 802.16m evaluation methodology, IEEE 802.16 Standard (December 2007)

QShine 2009

**Invited Session II – Overlay,
P2P Networks and Service
Oriented Architectures**

A Topologically-Aware Overlay Tree for Efficient and Low-Latency Media Streaming

Paris Carbone¹ and Vana Kalogeraki^{1,2}

¹ Department of Informatics, Athens University of Economics and Business

² Department of Computer Science and Engineering, University of California-Riverside

Abstract. Streaming a live music concert over the Internet is a challenging task as it requires real-time, high-quality data delivery over a large number of geographically distributed nodes. In this paper we propose MusiCast, a real-time peer-to-peer multicast system for streaming midi events and compressed audio data. We present a scalable and distributed tree construction algorithm where nodes across the Internet self-organize into a low-latency tree. Our system is built on top of the pastry DHT and takes advantage of the DHT's properties to construct an end-to-end low-latency dissemination tree using topology oriented information. The benefit of our scheme is that it is completely decentralized, allowing nodes to connect to each other using local information only, and achieves good performance by considering latency information when constructing the tree. Our experimental results illustrate the benefits of our approach.

Keywords: Overlay Networks, Multimedia Streaming.

1 Introduction

In the past few years, peer-to-peer multicast services have received a growing acceptance over traditional methods such as IP multicast that has been the de facto mechanism for delivering data streams to a large number of participants. Peer-to-peer systems offer a number of attractive characteristics including adaptivity, scalability and robustness, properties of increased importance with the growing popularity of the Internet today and the increasing interest for online multimedia content distribution.

However, multimedia streaming brings a number of challenges to the design of peer-to-peer multicast systems: First, multimedia data streams are produced in large volumes and high rates by a small set of sources to a large number of receivers. For these applications, low-latency delivery of the streaming data is of paramount importance. Second, such an application layer multicast system consists of a number of nodes that are geographically distributed, thus, the multicast system must consider not only the delay but also the topology of the nodes and their geographic proximity. Third, computational and communication resources are shared by multiple concurrent and competing streams; this poses certain restrictions on the number of connections of the peers.

A number of multicast streaming systems have been proposed in the literature. Most systems have made significant contributions on load balance and content distribution [3,5,6,9], while there are also a few examples [8] targeting latency requirements and low cost tree construction. End-to-end latency and cost measurements require relative topological approximations and frequently the combination of different optimal structures (min cost, min path trees, meshes etc.). There are also many challenges that need to be faced when dealing with delay metrics such as peer churn, node failures and dynamically changing application requirements.

In this paper we present MusiCast, a topologically-aware peer-to-peer multicast system. MusiCast builds upon the idea of making possible a live music concert, consisting of nodes with different roles (*i.e.* musicians, spectator nodes) to cooperate and transmit music over the Internet. The key challenge in the design of MusiCast is to construct a tree structure that distributes the load across all participating nodes and achieves this in a decentralized and scalable manner. We present a distributed tree construction algorithm where nodes organize into a topologically-aware, low-latency overlay tree. MusiCast is built on top of the Pastry DHT [15] taking advantage of its properties. The advantage of our scheme is that it achieves high data delivery ratios and low end-to-end latencies. MusiCast offers robustness to node failures and disconnections; thus, the failure of a node does not affect the performance of the rest of the system. We have implemented MusiCast on a local area testbed and evaluated its performance on various metrics including end-to-end delay, jitter and bandwidth used. Our experimental results demonstrate the efficiency and performance of our approach.

The rest of the paper is organized as follows. Section 2 presents our system model and overview of our approach. In section 3 we describe the system components, the tree construction algorithm, the run-time operation of our algorithm and our approach to failure recovery. In section 4 we describe our performance evaluation. Section 5 presents related work and section 6 concludes the paper.

2 Problem Formulation

In this section we first present our system model and then we give an overview of our approach.

2.1 Our System Model

The overall system consists of a set of overlay nodes N , divided into two different categories. Each node category represents a different layer of quality requirements and constraints that need to be taken into account by the system.

- $M \subset N$: Musician nodes are the main data sources of the system and are responsible for streaming midi or audio data. The number of musician nodes is typically small ($\|M\| = [1, 10]$), following a typical music band size. The main requirement of the M nodes is to maintain low playback latency to synchronize among themselves in order to achieve continuous and smooth delivery of the data streams.

Table 1. Notations explanation

Notation	Meaning
V	A set of all participating spectator nodes in the system
J	A set of nodes currently joined in the multicast tree
K	Candidate parents set, $k_i \in K : k_i \in J$ and $S(k_i) > 0$
v_i	A node $v_i \in V$
$p(v_i)$	The parent node of v_i on the multicast tree
$C(v_i)$	The set of children of v_i on the multicast tree
$d(v_i, v_j)$	distance between nodes v_i and v_j
$l(v_i)$	Tree path distance v_i from the tree root c
B	Bandwidth required by stream
$r(v_i)$	outgoing bandwidth of v_i
$S(v_i)$	The number of available slots in v_i

- $S \subset N$: Spectator nodes receive, forward and playback data streams as they are generated by the sources. The number of Spectator nodes can range from a few hundred to a few thousand nodes. Spectators' demands are the most challenging ones due to the scale of the spectators' subsystem. Their goals include: low delay in tree construction, small latency, minimum jitter and load balancing.

One of the Spectator nodes takes the role of the Coordinator c , which is responsible for musicians' synchronization, main sequence composition and tree construction initialization control. The coordinator also serves as the root of the multicast tree.

We assume that the overlay network of the spectators is represented as a graph $G = (V, E)$, where V is the set of all the spectator nodes, including the coordinator and $E = V \times V$ is the set of the edges between the nodes. The weight of each edge $\langle u_i, u_j \rangle$ represents the distance $d(u_i, u_j)$ between the two nodes. We will assume here that $d(u_i, u_j)$ denotes the actual unicast delay between u_i and u_j as the distance metric but it could also be replaced by other metrics. We assume that each node u_i has a maximum number of connections, also referred to as slots of u_i , $S(u_i)$. The maximum number of slots each node can handle is $\frac{r(u_i)}{B}$, where $r(u_i)$ describes the maximum outgoing bandwidth of u_i and B specifies the bitrate of the overall streaming sequence. Every node should offer at least one slot in order to join the system, so $S(u_i) > 0$. In order to disseminate a live data stream to all spectator nodes effectively, it is required to construct a spanning tree on G in which: (a) the degree constraints are satisfied and (b) the maximum end-to-end delay from the root to each node is minimized.

One important question is how to measure the distance between the nodes. We use a distributed distance measurement scheme based on the binning scheme proposed in [11]. The disadvantage of using other approaches where the distance between each pair of nodes is obtained using active end-to-end measurements, is that, the system would not be scalable due to the increased number of measurements needed ($O(n^2)$) and the lack of network bandwidth required for such

consumable operations. We note, however, that this is an NP-hard problem, however, our approach manages to calculate the distance between the nodes efficiently, as we will explain later in the paper.

2.2 Approach Overview

Our approach is two-fold: (a) First, at an initialization phase, our goal is, given a number of musician and spectator nodes, to construct a low-latency tree structure that offers small latency and a good load balance while respecting the network constraints at the nodes. The advantage of our approach compared to past techniques such as the ones proposed in Bullet, MeshTree and Coolstreaming [23,8], is that they often start by building random trees and then using an increasing amount of messages in a greedy order, aim to transform it into a low-delay tree. (b) Second, at the run-time phase, we employ optimization techniques in order to reduce the average delay to respond to dynamic changes to peer churn and resource availability. This will enable us to reduce unwanted jitter caused by joining or departing nodes that can affect the playback quality throughout the entire tree.

3 System Overview

In this section we discuss the operation of our system. MusiCast consists of musician and spectator nodes. We first describe the operation of the musician and spectator subsystems. Then we discuss our distributed distance measurement scheme and how topological awareness is accomplished using content stored at the peers, followed by the description of the tree construction algorithm used, our run-time optimizations and how we deal with failures. The architecture of our system is illustrated in Figure 1.

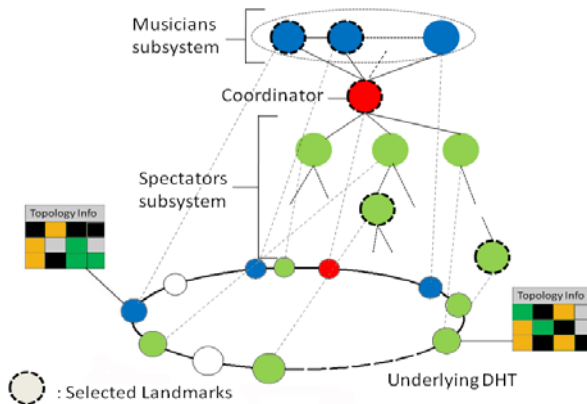


Fig. 1. Architecture of our System

3.1 The Musicians Subsystem

Each musician is responsible for streaming on a specific midi or audio channel while reproducing the sequences received from other musicians in real time. There are strict latency constraints that need to be taken into account. Event messages generated by each musician need to get to all other musicians (and to the coordinator) as soon as possible, without any extra delays, in order to achieve a responsive and effective playback. That is why we chose to establish a broadcasting scheme between them using the many-to-many unicast technique. Our goal is to keep low latencies, as large latencies could cause confusion to musicians, thus lowering the quality of their performance. Also, the number of musicians is ordinarily small and so as the bit rate of compressed audio and midi packets, so this technique is suitable for this specific setting. All midi packets received on each musician are instantly directed for playback without further buffering while compressed audio packets are buffered to the lowest degree. Control messages and basic synchronization between musicians are handled by the coordinator node. By synchronization we mean the maintenance of a global accurate timing, rate and channel distribution between them. The coordinator gives to each musician its global timing offset at startup with the use of the NTPv4 protocol which is shown to achieve an accuracy of 1-2ms in a LAN infrastructure or 8-10ms in a WAN. It is also the coordinator which instructs the musicians about the exact global time each one will start streaming on its specifically given channel. Each sequence stream from the musicians is also forwarded to the coordinator node which synthesizes the main sequence by aggregating all packets on top of the NTP timing protocol. This main sequence is being disseminated to the spectators' multicast tree from there.

3.2 The Spectators Subsystem

The spectators' subsystem consists of the spectator nodes and the coordinator which serves as the main source. Our goal is to organize the spectator nodes into a low latency multicast tree and achieve minimum overhead for control and optimizations.

The initial tree structure is an important decision, because despite optimizations, a reliable system is ought to guarantee high quality services from the point it begins operating. Many popular recent end-to-end multicast systems such as Bullet and MeshTree often start by creating a random structure [2,3,6,11]. Random tree structures offer some benefits such as resilience and costless initialization, though this could result in unbalanced situations with high average latency and jitter. Another disadvantage concerning these techniques is the over increasing overhead caused during runtime due to required optimizations which can affect playback quality greatly. A low cost initial tree on the other hand could guarantee a low average jitter while getting transformed easier into a more balanced one by following simple transformations as we will explain later.

Next, we will introduce some good attributes a dissemination tree is reasonable to have for high quality streaming. During a top down distributed construction of

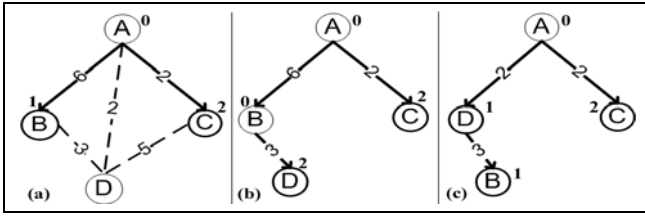


Fig. 2. Example: Operation of our System

the multicast tree there is a set of nodes J , $J \subset V$, that have already joined the overlay tree. When a node finds an available parent and establishes a connection it is considered as joined. Due to variable bandwidth availability, as it has been already mentioned in section 2.2, each node u_i is capable of supporting a certain fixed number of children (or outgoing connections) $S(u_i)$, known as slots. That signifies that there's only a set of candidate parents K at any time, specified as tree members having free slots.

Definition 1. Candidate parents K , $K \subseteq J, u_i \in K$ iff $u_i \in J$ and $S(u_i) > 0$.

Figure 2 shows an example with four nodes A, B, C and D for which the edges between them illustrate their distance (unicast latency in our example). Nodes A, B and C have already joined the multicast tree, so the set of joined nodes is $J = \{A, B, C\}$. Assume that their slots availability is as follows: $S(A) = 0, S(B) = 1$, and $S(C) = 2$. That denotes that if D considers joining the tree, it has to take into account only the set of candidate parents $K = \{B, C\}$ once A does not have any available slots left. Nodes B and C are children of A in the multicast tree and this set is defined as $C(A) = \{B, C\}$. Also the parent of a node u_i is defined as $p(u_i)$, so $p(B) = p(C) = A$. The problem here is which candidate D should choose as a parent. If D choose B as a parent then $d(D, B) = 3$ and $l(D) = 9$, where $l(u_i)$ is the tree path distance of a node u_i . Alternatively, by joining C , $d(D, C) = 5$ and $l(D) = 6$, so the choice that should be made here is non-trivial. Although both per-hop and end-to-end latency attributes can be considered for high quality streaming, we chose to use the per-hop distance as priority constraint. In the example that means that $p(D)$ should connect to B despite high end-to-end latency. Formally, the parent of each node following that pattern should be found using the following min-cost rule:

Definition 2. $p(u_i) = u_j$ iff $d(u_i, u_j) == \min\{d(u_i, u_k)\}, \forall u_k \in K$.

A tree constructed following the min-cost join metric defined above is essential for a starting point on multicast streaming and offers the minimum possible per-hop latency which is important to maintain low jitter levels at startup. However, this is not sufficient for scalability and overall quality. Another key metric that should be taken into account is the average end-to-end latency from the root c to each tree node, defined as $\sum_{i=0}^{\|J\|} \frac{l(u_i)}{\|J\|}$. It is quite simple to transform a min

cost tree into a low latency one by applying the following rule to each node until it gets to the highest possible level in the tree:

Definition 3. *if $p(u_i) == u_j$ and $d(u_i, p(u_j)) < d(u_j, p(u_j))$ then $p(u_i) \rightarrow p(u_j)$ and $p(u_j) \rightarrow u_i, \forall u_i \in J$ where $S(u_i) > 0$*

A node occasionally checks whether it is closer to its grandparent than its parent and if so it can be swapped with its parent. This is needed because of high churn: in a dynamic environment a peer can connect and disconnect from the tree at random times and without a priori notification. This can leave the tree unbalanced. Each node having that property moves closer to the root and after a period of convergence the resulting tree has nodes relatively close to the root occupying the top levels of the tree while nodes further from the root occupying the bottom levels. That is an effective way for minimizing the average end-to-end delay. In the previous example, D should swap with its parent B once $d(A, D) < d(B, D)$ resulting in $d(A, D) = l(D) = 3$ and a decreased average latency of 3. In our system, the low cost tree is being created during a pilot initialization phase, in which topology information is being collected and then the optimization process takes place at run-time.

3.3 Accomplishing Topology Awareness

One important challenge in our setting is whether it is possible to gather topological information in a manner that is both practical and scalable and if so, how could this information be effectively incorporated into the design of distributed systems such as overlay networks and content distribution systems. An effective solution was proposed in [11] called *binning scheme*, in which technique nearby nodes cluster themselves into groups, called 'bins', such that nodes that fall within a given bin are relatively close to one another in terms of network latency.

The binning scheme is fully distributed and it requires only a set of k well-known machines l_1, l_2, \dots, l_k to be set as landmarks in a specific ordering. The technique works as follows: a node measures its distance, i.e. round-trip time, to this set of well known landmarks and independently falls into a particular bin based on these measurements. This is performed by all the nodes in the network. We chose to use this technique because it offers great advantages: (a) it is simple and cost-effective, there are only $O(nk)$ operations required, where k is the number of landmarks and n the number of all nodes, instead of $O(n^2)$ and (b) it requires very little support from the infrastructure. The only infrastructure required is a small number (depending on the overlay size, usually 4-6 suffice) of relatively stable landmark machines which they only need to echo 'ping' messages. These landmarks could in fact be unsuspecting participants in the binning scheme. Landmarks do not actively initiate measurements nor gather or disseminate measurement information. Another advantage of the binning scheme is that it is scalable because nodes independently discover their bins without communicating or coordinating with other application nodes. Finally, this technique is robust to the failure of one or more landmark nodes as described in section 3.7.

In our approach we have extended the binning scheme as follows: The bin a node v_i belongs to is represented as a vector of values in specific ordering $\langle q_1, q_2, q_3, q_4 \rangle$, where q_i is a certain level of latency between landmark l_i and v_i . Levels of latency are usually between 3 to 5 and are computed by the landmarks based on ping measurements gathered at the system's initialization period, set in a way to dissociate nodes most effectively. We have modified the original binning scheme by setting the order of the participant landmarks due to their distance from the tree root and having as first landmark the actual root of the tree. The distance metric we used to approximate the distance $d(B_i, B_j)$ between two different bins, B_i and B_j , is the following:

$$d(B_i, B_j) = \sum_{l=0}^k [\|B_{il} - B_{jl}\| * (k - l + 1)] \quad (1)$$

For example the distance between the bins $B1 = \{2, 2, 1, 0\}$ and $B2 = \{1, 2, 2, 2\}$ is $(4 + 0 + 2 + 2) = 8$ based on the metric mentioned above. As the distance metric dictates, landmarks that are closer to the root are more important than landmarks further from it so along with relative distance, a bin also reflects the actual distance from the tree root in relation to other bins.

3.4 Content Management

Topological information in our system is stored on specific responsible nodes. There are two types of topological information. First we have the *bin data* which specifies close nodes in the overlay due to the unicast latency proximity metric. Second, we have the *zone* oriented content. A zone is one extra layer of topology measure which is derived directly from the bins. Each zone contains all bins starting with this zone which is simply the first value of a bin vector (always referring to the system's root). For example zone '0' contains all bins starting with '0' (eg $\langle 0, 2, 2, 3, 2 \rangle$, $\langle 0, 1, 2, 0, 3 \rangle$). By using zones, we can reduce the amount and size of topology oriented messages by requesting bins on a specific zone to apply operations and not involving the remaining bins in the system. We can also guarantee some scalability during tree construction by starting the top-down building algorithm from member nodes of bins belonging to zone 0 and then continuing to the next zones.

We have built our system on top of the Pastry DHT [15]. The Pastry DHT is mainly used for storing the content management information. The advantage of Pastry is that the content storage is well-balanced across the system, and retrievals of it can be achieved only with a small amount of messages. Responsible content nodes, maintain certain states of the bins and zones and update them based on any changes that occur in the tree. Bin data state for example specifies whether all bin members have joined the tree. When all nodes on a bin have joined the multicast tree this is reflected at the zone state too so that everyone knows whether all zone bins have joined the tree when asking for zone specific information. When a content state changes, the responsible node for the specific content (having the numerically closest ID hash number to the content's hash

Algorithm 1. Tree Construction Join Algorithm()

```

1:  $SLOTS \leftarrow node.availableSlots$ ;
2:  $currentZone \leftarrow node.zone$ ;
3: if ( $SLOTS > 0$ ) then
4:   obtain bin information;
5:   if (unjoined node on same bin  $> 0$ ) then
6:     add unjoined nodes in  $C(node)$ 
7:   end if
8:   while ( $SLOTS > 0$ )AND( $currentZone \leq MAXzone$ ) do
9:     // there are still slots available
10:     $bins[] \leftarrow zone[currentZone + +].bins$ ;
11:    sort( $bins[]$ ); //due to the distance from current node's bin
12:    for all (bins in  $bins[]$ ) do
13:      add the closest not joined node in  $C(node)$ 
14:      if ( $SLOTS == 0$ ) then
15:        return;
16:      end if
17:    end for
18:  end while
19: end if
20: return;

```

number), is being notified to update its content. Content can also be replicated to multiple responsible nodes thus making the system more robust during node failures. Information retrieval is pretty straightforward, by using the lookup(ID) function on the DHT to get the information needed by having only a given ID. Note that all DHT operations in Pastry require $O(\log n)$ messages.

3.5 Initial Tree Construction

During the initialization phase all musician nodes join the overlay apart from the spectator nodes, the binning information is being computed and then stored into the DHT and the low cost tree is formed for the streaming process to begin. At first, the coordinator and then the musicians join the system and a phase begins in which the number of spectators join the system and compute their distances to binning landmarks which have been predefined by the root. The first landmark is always the root and the next landmarks are usually occupied by the musicians considering their stability in addition to more, possibly random landmarks, all ordered due to their distance to the root. After enough latency measurements have been made on all landmarks, the bin levels are computed on each landmark and broadcasted to the whole system using Scribe [5], an event-based notification system built on top of Pastry. The latency range of each level is chosen based on the variance, the mean latency value and the number of latency levels (in our experiments we used 4 zones). For example, using four zones, latencies are normalized and dissociated into ranges by the z values of the latency distribution measured $z_i = (\mu + i * \sigma)$ as such $\langle [0, z_{-1}], [z_{-1}, z_0], [z_0, z_1], [z_1, +\infty) \rangle$

thus granting the maximum separability possible with the use of bins. When all bins are computed and content updates finish this phase is over and the initial tree construction phase begins. The initial tree construction is initiated by the coordinator root which runs first the initial low-cost tree building algorithm (Algorithm-1). The resulted tree is also end-to-end-latency aware due to zone priorities used in the join process.

The algorithm works as follows: when a parent finds an appropriate node to attach to its children, the new child starts the same process and searches for appropriate children. It begins by getting its own bin's content information and then checking whether there are any unregistered nodes and if so, it asks them to become its children. If there are more slots available it continues the search for child nodes by asking the DHT for its own zone information. If there are bins containing unregistered nodes it asks those nodes to become its children in a specific order, relevant to the distance to those bins. If the zone does not contain any bins with unregistered nodes it asks the DHT for the next zone information and the algorithm continues until there are no available slots at this node or when all nodes have been registered during the initialization phase. Note that after all nodes in the overlay have joined the multicast tree, the coordinator orders the musicians to start streaming their sequences on a specific global time point and the stream is then being disseminated in the multicast tree, setting the outset of the run-time phase.

We will demonstrate the usage of Algorithm-1 by giving an example. In figure 3(a) we visualize a part of the multicast tree during the initialization phase (top down initial construction). Nodes *A* and *B* have already joined the tree while *C*, *D* and *E* wait to get placed on appropriate positions. Next to each node we've attached the actual bin ID for its corresponding bin. In (a) node *B* searches for one more node to add to its children by following Algorithm-1. Unregistered nodes (*C*, *D*, *E*) have been grouped by their zone below the graph. Node *B* first asks for zone 2 information once it belongs to zone 2 (and there are no unregistered nodes in its own bin) and gets only the bin $\langle 2122 \rangle$ in which contains the node *C*. Then, *B* asks *C* to join the tree as one of its children and stops there because it has no slots available. Node *C* now (figure 3 (b)) finds no unregistered nodes in its own bin and also gets informed that zone 2 bin nodes have all been registered. At that point node *C* can take the zone 3 list of unregistered bins, which are $\langle 3101 \rangle$ and $\langle 3231 \rangle$ which contain nodes *D* and *E* respectively. Distances due to the relative bin metric proposed in section 3.3 of this paper, are $d(\langle 2122 \rangle, \langle 3101 \rangle) = 9$ and $d(\langle 2122 \rangle, \langle 3231 \rangle) = 10$ so *C* first adds node *D* to its children. Assuming that *C* maintains one more available slots, it adds node *E* too as its second child and stops there (figure 3(c)).

3.6 Run-Time Optimizations

After the initial phase, during run-time, the tree is being transformed into a more balanced one with the use of tree transformations. We followed the optimization rule (3) stated in section 3.2 (Definition 3). Occasionally each registered node in the multicast tree (or a newly joined node) checks whether it is closer to its

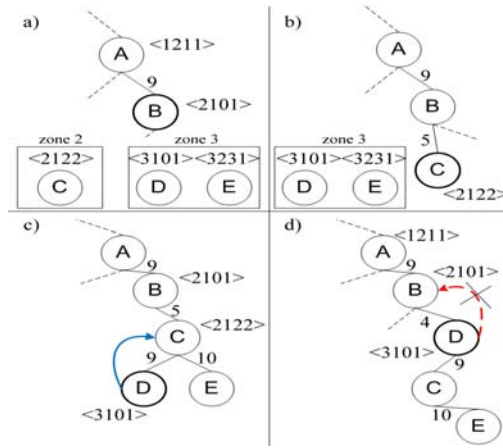


Fig. 3. Examples of the initialization and optimization phases

grandparent than its parent is, due to the relative distance. If that is true and the node has at least one available slot, its parent becomes its child and it takes the place of its parent in the tree by setting its grandparent as its parent. Then, it checks again for a new swapping with its new parent. By following this rule, nodes that are closer to the root tend to ascent the tree and thus granting a low average end-to-end latency for the system.

In our previous example (figure 3 (c)), node D checks whether it is closer to B than C is, by calculating the relative distances $d(D,B)=d(\langle 3101 \rangle, \langle 2101 \rangle)=4$ and $d(C,B) = d(\langle 2122 \rangle, \langle 2101 \rangle)=5$. It is obvious that it can be swapped with its parent (assuming that C maintains an additional free slot) and so its new parent now is B (figure 4 (d)). However, $d(D,A) = 13$ while $d(B,A) = 9$ so D cannot proceed on further optimizations for the time being.

3.7 Failure Recovery

There is a number of possible failures that can happen during the system’s operation: (a) spectator node failures, (b) failure of a landmark node, and (c) failure of musician nodes. The easiest to deal with is spectator failures. If a spectator leaves the system without warning, its child in the multicast tree will diagnose its parent’s loss and will instantly ask it’s grandparent to add it to its children. If that cannot happen the child asks the next closest node from the same bin to add it until it finds an available parent. The second type of failure which is more serious, is the failure of a landmark node. In this case, considering that only a small number of landmark nodes can concurrently fail, the best solution is for each node to drop the landmark identifier from its bin vector and new responsible landmark nodes need to be found by Pastry. This will require only $O(\log n)$ extra messages for each bin to find the new responsible node via Pastry. Finally, the failure of a musician node could result in the loss

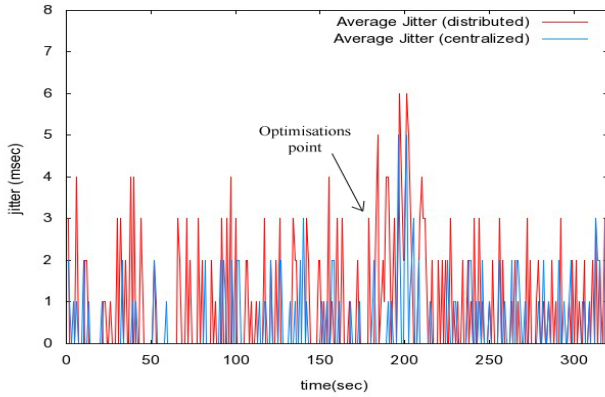


Fig. 4. Average Jitter Measurements experienced by the Spectator nodes

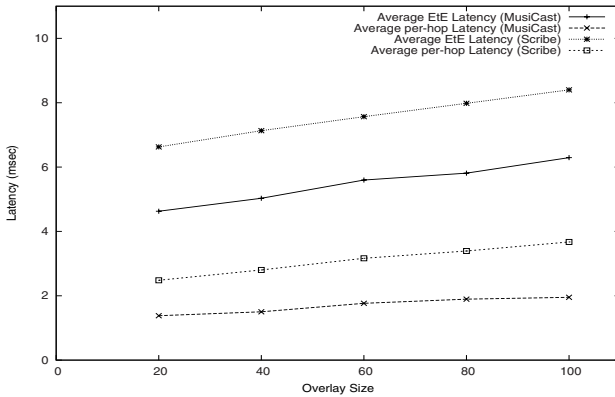


Fig. 5. Latency as a function of the Overlay Size

of one channel but not the failure of the whole system. Even if the root failed, the closest spectator or musician could be instantly act as the new tree root by broadcasting a control message of its address to the entire overlay indentifying itself as the new root.

4 Performance Evaluation

We have performed a series of experiments to evaluate the efficiency and performance of our system, using up to 100 peers deployed in a network of 20 local x86 machines consisted of a 3.0GHz single core CPU and 1024MB RAM which were connected via 100Mbps Fast Ethernet.

Three of the nodes acted as musicians of the system (sources) and the coordinator which was running at a well known address acted as a bootstrap for the

Pastry DHT. As a typical streaming content we chose a combination of midi sequences among a 64kbps audio sequence, taken from typical big band tracks. We tried to select music having occasional variations in rate in order to observe our system's response in rapid rate peaks. For delay oriented measurements, we have managed to achieve microsecond accuracy to comply with the LAN's typical latency scale having a 8-10 microseconds possible amount of error. Also, all sequences were streamed via UDP for the minimum possible delay.

To evaluate the performance of our system, we have compared it with a centralized version of it. This centralized version uses the same algorithm (Algorithm-1) introduced for top-down tree construction during the initialization phase, with the difference that a central process manages the whole tree construction and orders all nodes based on their distance from the root before connecting each of them to the appropriate low cost parent. That results in a more balanced low cost tree which maintains a low end-to-end latency and better load balance from startup. After the initialization phase, during run-time, the centralized system continues to operate normally as in the distributed scheme. We have also evaluated our system in comparison with an implementation of Scribe, which is another popular multicast system on top of Pastry [5]. Scribe creates a multicast tree at runtime by using reverse routing paths to a specific node (known as rendezvous point or group creator). In this implementation the coordinator is the actual scribe multicast group's creator and we've also included an extra check during initialization of each node on bandwidth availability as follows: when a Scribe node tries to establish a route to the multicast tree root, it checks whether the next hop (parent) following this route has available slots, if not it rejoins the DHT using a different random node ID, thus connecting to the group from a possibly different location. This process continues until the node finds a parent with available slots.

In the first set of experiments we measured the average jitter experienced by the Spectator nodes. The average jitter level among the spectators is an important consideration because it affects playback quality directly, especially if it outruns a predicted buffering delay. Figure 4 shows the average latency of the Spectator nodes. During our experiments we noticed slight increases in jitter levels (typically 0-4 milliseconds in a LAN) on specific parts of a music act. These jitter peaks can be noticed in Figure 4 among the distributed and the centralized version of MusiCast. Even though the peaks are unnoticeable due to their microsecond level in a WAN infrastructure, jitter could cause increased variations in length resulting to decreased playback quality. There are two types of jitter peaks that can be noticed in Figure 4 in both versions: (a) some casual small peaks that appear every 65 seconds, starting from second 48 and (b) a larger peak at second 200 having increased spanning. Small casual peaks appear due to increased bit rate at specific parts of a music act and that explains their repetitive nature. The bigger peak on the other hand is caused by the increased amount of tree operations during the optimization period. It is clear that the centralized scheme has lower average jitter before the optimizations occur and the same jitter levels after them. That implies that low end-to-end latency

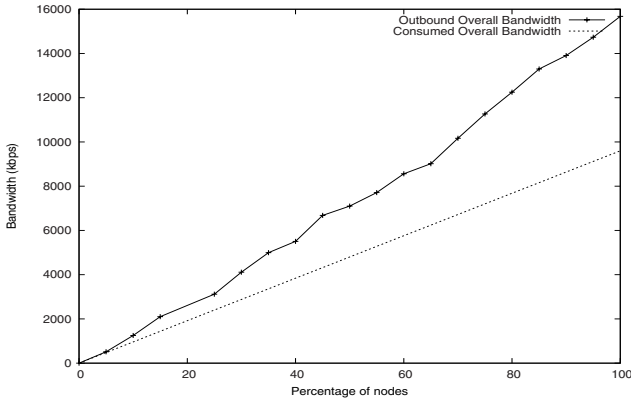


Fig. 6. Overall Bandwidth

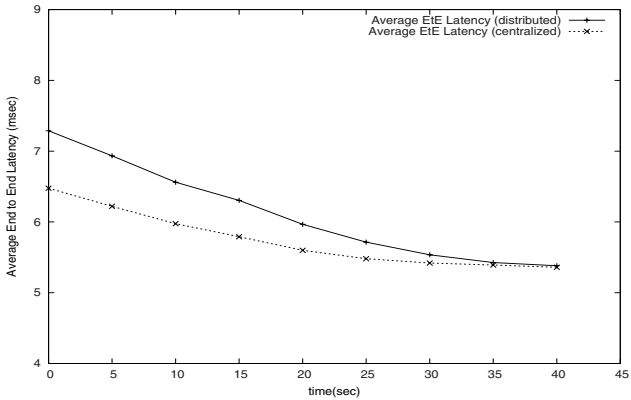


Fig. 7. End-to-end delay during optimizations

decreases the average jitter of the system and that is reasonable concerning shorter paths to the root and possibly fewer connections.

Our system’s low latency performance can be seen in Figure 5. The latency is shown as a function of the overlay size in comparison with the one of the special version of Scribe described above. Our algorithm achieves 27% improvement on the average end-to-end latency while showing similarly low increase rate to Scribe. Per-hop latency is also lower in our system due to the initial tree values that have been maintained.

Another metric that needs to be considered in every multicast system is the overall offered and consumed bandwidth. There must always be enough remaining bandwidth for new nodes to join the system at any time. In Figure 6 it is clear that the available bandwidth is over-increasing during the joining of new

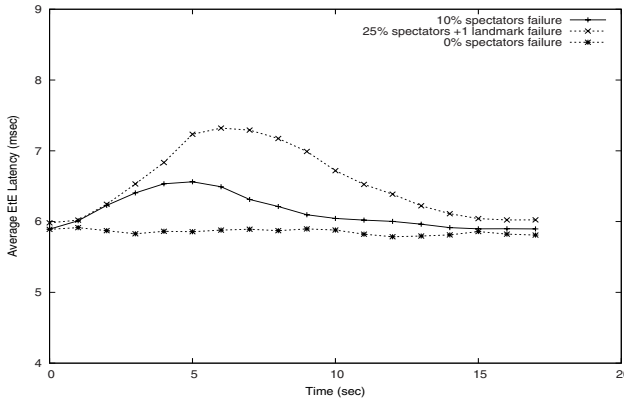


Fig. 8. Latency during failures

nodes. The available bandwidth cannot reach zero because once we've set the lower limit of one slot to each node, the actual increase of the offered bandwidth can be at least equal to the one of the consumed bandwidth. Also, the joining process is flexible enough, allowing a new node to join the closest available neighbor easily without imposing it like scribe does when sticking only to a specific route ordered by Pastry.

System's convergence during optimizations lasted for about 35 seconds on the distributed version and 25 seconds on the centralized scheme because the tree was already balanced in some degree. In Figure 7 the average end-to-end latency decrease is visualized during the optimization period. Both schemes seem to converge nearly on the same optimal average end to end latency level. We can conclude here that at first the centralized scheme is having a better structure, though after the optimization period the distributed scheme managed to achieve the same streaming quality level with the centralized one.

We have finally tested our system's recovery performance during different failure rates. Latency measurements took place throughout the system from the moment a number of randomly selected spectators had failed. During this period the recovery procedure took place which was described briefly in section 3.7 . As the figure shows, the system managed to achieve a complete recovery when 10% of the spectators had failed, in about 13 seconds. However, recovery was a more difficult task when 25% of the spectators left the system including one landmark. As shown in figure 8, in this case, the system managed to converge on a little higher point of average latency after 15 seconds and that's because the binning scheme was a little less accurate than before, having one less landmark in each bin. In special cases when more landmarks could fail, new landmarks need to be chosen again resulting in a small period of reconstruction ending in a complete recovery of the system, otherwise, continuing with the reduced bins is preferable and cost effective.

5 Related Work

Several recent projects make use of application-level multicast and overlays to achieve media streaming [2,3,5,6,8,9,11,12]. Some of them such as Scribe, ChunkySpread and SplitStream have adopted tree or multi-tree structures achieving low overhead solutions although they have failed on maintaining low end-to-end latency or link stress and effective failure recovery [5,6,9]. Bullet, MeshTree, mTreeBone and Coolstreaming offer a different approach by including an initialization using a random structure followed by a series of optimizations during runtime. These solutions have many benefits when streaming encoded pre-buffered content or other content not sensible by high jitter. They also offer resilience and simplicity during initialization. However these techniques are not suitable for streaming content sensible by jitter such as midi events and that's due to their: a) initial high delay structure and b) the increased overhead caused during runtime to achieve all the required optimizations.

Mesh structures are more preferable to trees in projects such as Bullet and CoolStreaming and that's because meshes offer more robustness and ease of locating and maintaining low latency links between peers while achieving a good load balance [2,3,12]. However, meshes can potentially incur high network or CPU overheads due to the demand of extra amount of control messages for mesh maintenance. In our system meshes were not suitable because control messages could make an impact to jitter values and as a result the playback of midi messages could be affected. There is also a hybrid approach, combining tree and mesh structures such as the one on MeshTree and mTreebone where trees and meshes' favorable properties have been merged achieving an impressive result [8,11]. Hybrid solutions are essential for high-volume, bandwidth-demanding data streaming yet are not preferable for medium-volume data such as midi events or compressed live audio types on which the extra overhead caused by the meshes' control messages used could even outmatch the actual data size and cause undesirable jitter.

Finally none of the systems mentioned has achieved topology awareness in the degree our system did. We have managed to replace any actual delay computation between overlay nodes, with the distance metric of the binning scheme [1] while achieving the minimum possible overhead to establish such an accurate delay evaluation scheme without even measuring corresponding delays by taking advantage of the content storing properties of a DHT such as Pastry.

6 Conclusions

In this paper we have presented the design principles and mechanism behind MusiCast, a peer-to-peer multicast system specifically oriented to low and medium size content streaming, such as midi events and compressed audio. Our system manages to synchronize musician nodes that produce midi events or audio streams and the overall stream is then being passed to a large number of connected spectators in a highly scalable way by building a topology awareness

scheme on top of the Pastry DHT, while keeping low latency and low average jitter levels. Our results extracted from various local experiments certify the effectiveness of our approach in dealing with diverse latencies and node capacities.

Acknowledgements

This research has been supported by NSF Award 0627191 and a Marie-Curie fellowship. We would like to thank A. Rowstron and P. Druschel for their open source implementations of Pastry (Free Pastry).

References

1. Ratnasamy, S., Handley, M., Karp, R., Shenker, S.: Topologically-Aware Overlay Construction and Server Selection. In: IEEE INFOCOM (2002)
2. Kostic, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh. In: SOSP 2003 (2003)
3. Zhang, X., Liuy, J., Liz, B., Yum, T.-S.P.: CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming. In: IEEE INFOCOM, Miami (March 2005)
4. Magharei, N., Rejaie, R., Guo, Y.: Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches. In: IEEE INFOCOM (2007)
5. Castro, M., Druschel, P., Kermarrec, A.-M., Rowstron, A.: Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications* 20(8) (October 2002)
6. Venkataraman, V., Francis, P., Calandrino, J.: Chunkyspread: Multi-tree Unstructured Peer-to-Peer Multicast IPTPS, Santa Barbara, CA (2006)
7. Jin, X., Xia, Q., Gary Chan, S.-H.: A Cost-based Evaluation of End-to-End Network Measurements in Overlay Multicast. In: IEEE INFOCOM (2007)
8. Tan, S.-W., Waters, G., Crawford, J.: MeshTree: A Delay-optimised Overlay Multicast Tree Building Protocol. University of Kent, Technical Report 5-05
9. Castro, M., Druschel, P., Kermarrec, A.-M., Nandi, A., Rowstron, A., Singh, A.: SplitStream: High-Bandwidth Multicast in Cooperative Environments. In: *ACM SIGOPS Operating Systems Review* (2003)
10. Chu, Y., Rao, S.G., Seshan, S., Zhang, H.: A Case for End System Multicast. In: *ACM Sigmetrics, Marina de Rel, CA* (2002)
11. Wang, F., Xiong, Y., Liu, J.: mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast. In: ICDCS, Ontario, Canada (2007)
12. Li, B., Xie, S., Qu, Y., Keung, G.Y., Lin, C., Liu, J., Zhang, X.: Inside the New Coolstreaming: Principles, Measurements and Performance Implications. In: IEEE INFOCOM, Phoenix, AZ (2008)
13. Banerjee, S., Lee, S., Bhattacharjee, B., Srinivasan, A., Zhang, X.: Resilient Multicast using Overlays. In: *ACM SIGMETRICS* (2003)
14. Banerjee, S., Kommareddy, C., Kar, K., Bhattacharjee, B., Khuller, S.: Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications. In: IEEE INFOCOM, San Fransisco, CA (2003)
15. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: *IFIP/ACM International Conference on Distributed Systems, Heidelberg, Germany* (2001)

Similarity Searching in Structured and Unstructured P2P Networks

Vlastislav Dohnal and Pavel Zezula

Faculty of Informatics, Masaryk University,
Botanicka 68a, 602 00 Brno, Czech Republic
{dohnal, zezula}@fi.muni.cz

Abstract. The exponential growth of digital data in contemporary computer networks induces a lot of scalability, resilience, and survivability issues. At the same time, the increasing complexity of digital data makes the task of similarity searching that is inherently difficult, more and more important. In this paper, we report on the Multi Feature Indexing Network, MUFIN, which is an extensible, scalable, and infrastructure independent similarity search engine. It is able to achieve high performance and guarantee quality of service by applying structured Peer-to-Peer networks. On the other hand, its unstructured version based on self-organizing principles is extremely robust and able to operate in very volatile environments. To exemplify MUFIN's properties, an on-line demo is available for public use.

Keywords: similarity searching, structured peer-to-peer network, unstructured peer-to-peer network, self-organizing system, metric space, scalability, resilience to failures, performance evaluation.

1 Introduction

Similarity is a central notion throughout human lives. In perception, the similarity between sets of visual or auditory stimuli influences the way in which they are grouped. In speech recognition, the similarity between different phonemes determines how confusable they are. In classification, the category of a new instance may be influenced by the similarity of the new instance to past instances or to a stored prototype. In memory, it has been suggested that retrieval of a cue depends on similarity of past memory traces to the representation of the cue. Since almost everything that we see, hear, read, write and measure is, or very soon will be, available in a digital form, computer systems must support similarity. But the growth of the amount of digital material distributed in large-scale wired and wireless networks is posing another big challenge. The exponential increase of data volume makes the performance, configuration, cross-layer approaches, scalability, resilience and survivability an important matter of concern. However, the core ability of future data processing systems should be developed around effective and efficient similarity management of very large and growing collections of data.

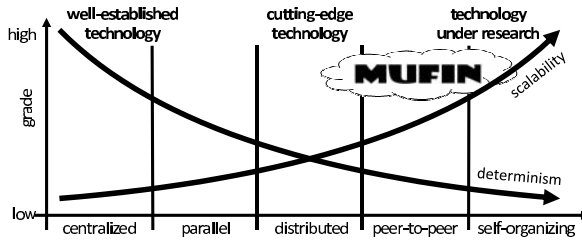


Fig. 1. Trade-off between scalability and determinism in system control

As Figure 1 outlines, we believe that a future search system will be created upon the divergence of scale and determinism. The scalability will be more and more important with respect to the data volume, number of users, query execution response time, number of different query types produced by digitization and content enrichment techniques, as well as the multi-modal approach to querying. On the other hand, the determinism in answering queries, i.e. providing always the same answers to the same queries, will be substituted by satisfactory results or even recommendations. Queries will also be much more personalized and influenced by context and executed on hardware most suited to the given workload. In any case, the exact match will be more and more often accompanied by extensive use of similarity searching.

In this paper, we present our Multi-Feature Indexing Network initiative and explain how our approach can contribute to the quality of service and robustness objectives of future similarity search systems. In particular, we give an architectural view of Multi-Feature Indexing Network in Section 2. Two instances of MUFIN defined as structured Peer-to-Peer networks are summarized in Section 3. Whereas, a system operating as an unstructured Peer-to-Peer system with self-organizing abilities is described in Section 4. Both these sections are accompanied with a sketch of experimental trials showing their properties. In Section 5, a demonstration application for image content-based retrieval is given. As for future applications, there are two examples of searching by biometric characteristics. The paper concludes in Section 6.

2 Multi-Feature Indexing Network

In this section, we present and demonstrate capabilities of the Multi-Feature Indexing Network, so-called MUFIN 1. From a general point of view as shown in Figure 2, the search problem has three dimensions: (i) data and query types, (ii) index structures and search algorithms, and (iii) infrastructure to run a system on. MUFIN adopts the *metric space* model of similarity. Its indexing and searching mechanisms are based on the concept of *structured* and *unstructured* Peer-to-Peer (P2P) networks, which makes the approach highly scalable and independent of the specific hardware infrastructure.

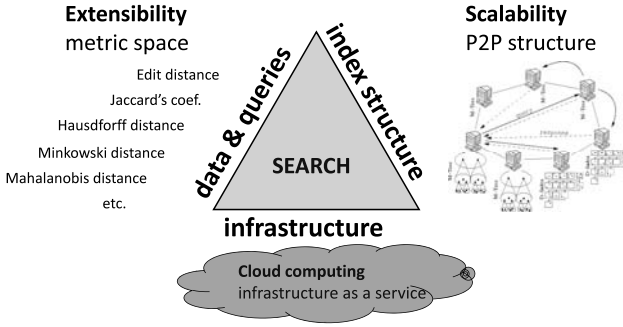


Fig. 2. Basic concept of MUFIN [2]

2.1 Modeling Similarity

The metric space model of similarity has already proved to be a very powerful concept for expressing many different forms of similarity of vectors, strings, sets and other data types. Most of the available technologies for processing metric data have been summarized in a recent book [3].

A *metric space* $\mathcal{M} = (\mathcal{D}, d)$ is defined for a *domain* of objects (or extracted features) \mathcal{D} and a total function d that evaluates *distance* between a pair of objects. The properties of this function are: non-negativity, symmetry and triangle inequality. The distance expresses *dissimilarity* between two objects. Examples of distance functions are L_p metrics (City-block (L_1) or Euclidean (L_2) distance), the edit distance, or the quadratic-form distance. Whereas examples of objects are a color histogram extracted from an image and stored as a vector, or a shape of hand expressed as a polygon.

There are two basic types of similarity queries: *range query* and *k-nearest neighbors query*. The range query $R(q, r)$ is specified by a query object $q \in \mathcal{D}$ and a query radius r . From a database $X \subset \mathcal{D}$, the query retrieves all objects found within the distance r from q . The definition is as follows:

$$R(q, r) = \{o \in X, d(o, q) \leq r\}.$$

Whenever we want to search for similar objects using a range search, we must specify the maximum distance for objects to qualify. But it can be difficult to specify it without some knowledge of the data and the distance function. An alternative way to search for similar objects is to use the k-nearest neighbor query $kNN(q)$. It retrieves the k nearest neighbors of the object q . Formally, the response set can be defined as follows:

$$kNN(q) = \{R \subseteq X, |R| = k \wedge \forall x \in R, y \in X - R : d(q, x) \leq d(q, y)\}.$$

2.2 Architecture

MUFIN, schematically depicted in Figure 3, has a four-tier architecture. The lowest tier is represented by a computer network and its hardware infrastructure

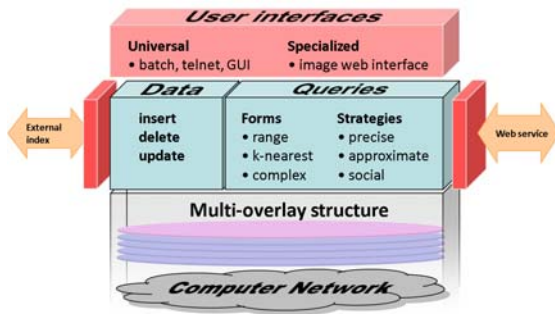


Fig. 3. Overview of MUFIN [2]

the system is running on. The executive core of MUFIN in the second tier is formed by several distributed indexing structures (*overlays*) that exploit the paradigm of P2P networks both in their structured and unstructured variants. Each of these overlays maintains data specific to it and distributes them among its (logical) peers. For example, an overlay can be defined for shape descriptors or color histograms in case of images, or protein spectra vectors in case of biological data. The number of logical peers in respective overlays and their mapping to physical computers are the main parameters that affect the system's searching performance.

From the third tier point of view, the logical peers of all overlays form a single virtual overlay with a uniform access to individual members. More precisely, the logical peers of different overlays mapped to the same physical host constitute a peer of this virtual overlay. The third tier provides interfaces for data maintenance (inserting and deleting data) and query specification, considering both the query form (range, nearest-neighbors or complex queries) as well as the strategy for query execution. Possible strategies are precise and approximate query evaluation. From the system point of view, these interfaces come in the form of a native API, a web service interface, or a plug-in interface for linking with external services.

Finally, the top-level tier represents interfaces allowing regular users to interact with the system. We have defined several general-purpose interfaces suitable for any application domain. However, they lack the comfort of a specialized interface. In Section 5.1, we present an example of an interface specific for image retrieval.

2.3 Properties

MUFIN is built by means of the Metric Similarity Search Implementation Framework (MESSIF) [4] – a large and extending Java library of metric searching implementation tools. It gives MUFIN flexibility in applying suitable implementation strategies for specific purposes and fast adoption of new progressive

solutions as they come from research. The properties of MUFIN can be summarized by the following attributes:

- Extensibility:** Different similarity search indexes for specific applications can be built with a single tool;
- Scalability:** Due to the underlying P2P technology, extremely large datasets can be processed;
- Adaptability:** In highly-volatile or unreliable environments, self-organizing principles can be implemented and the system can operate in unstructured P2P networks;
- Multi-modal Queries:** In order to adjust effectiveness of search according to needs of individual users, several overlays can be combined together using a monotonic aggregation function [5];
- Approximation:** To further improve performance, approximation techniques can be applied to query evaluation [6,7];
- Infrastructure Independence:** The networking module uses standard IP protocols. Each peer is identified only by its IP address and port number, so the mapping of the system to a hardware infrastructure is extremely flexible. For example, an instance of MUFIN can operate on a local network of common workstations, on a single multiprocessor machine, on a world-wide network, or even on a GRID system.

3 Structured Networks

In this section, we focus on indexing mechanisms of MUFIN that create purely-decentralized and structured P2P networks. In general, each peer of such a system consists of the following components and expects them from the other peers: (i) resources – storage and computational power, (ii) communication – a peer can contact any other peer directly if it knows its network identification, and (iii) navigation – internal structure that ensures correct routing among the peers. To ensure maximum scalability, the system also adopts requirements of the Scalable and Distributed Data Structures [8]: (i) data expands to new peers gracefully if and only if the peers already used are efficiently loaded, (ii) there is no master site to be accessed when searching for objects, e.g., there is no centralized directory, and (iii) the data access and maintenance primitives (search, insert, split, etc.) never require atomic updates to multiple peers.

In the following, we describe a space-partitioning technique called Generalized Hyperplane Tree Star and a space-transformation technique named Metric Chord.

3.1 GHT*

The Generalized Hyperplane Tree Star (GHT*) [9] is a decentralized structured P2P network that distributes data to peers based on the generalized hyperplane partitioning principle. Each peer maintains a tree structure called *Address Search Tree* (AST). An example of AST with the corresponding space partitioning is

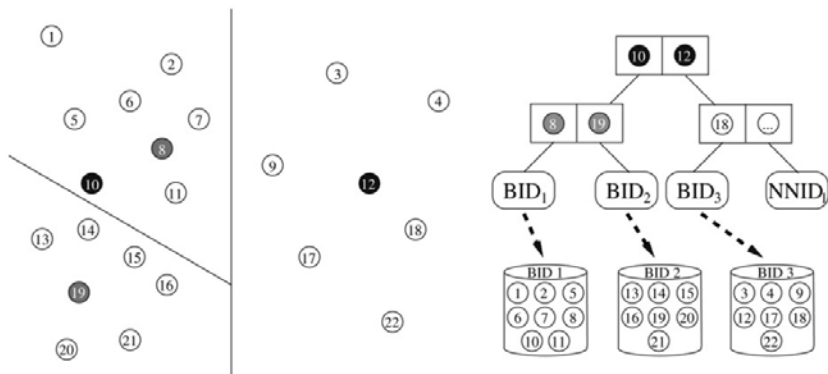


Fig. 4. Address Search Tree with the generalized hyperplane partitioning [9]

depicted in Figure 4. Internal nodes of AST store routing information – the definition of hyperplane. In metric spaces, the generalized hyperplane is defined using two objects p_1, p_2 , so-called pivots. The data objects $o \in \mathcal{M}$ such that $d(o, p_1) \leq d(o, p_2)$ form the left partition whereas the other objects form the right partition. Leaf nodes of AST store pointers to local buckets (denoted as BID) or to other peers (denoted as NNID). A bucket is a limited storage space dedicated for data objects, e.g., a memory segment or a disk block. The number of buckets managed by a peer depends on its own potential and capacity. Since the structure is dynamic and new objects can be inserted at any time, a bucket on a peer may reach its capacity limit. In this situation, a new bucket is created and objects are redistributed between these two buckets following the hyperplane newly defined. The new bucket may also be allocated on a different peer. Thus, the structure grows as new data come in.

The core of the algorithm lays down a mechanism for locating the respective peers that hold requested objects. Whenever a peer wants to query or modify the data, it must first consult its own AST to get locations, i.e. peers, where the data resides. Then, it contacts the peers via network communication to actually process the operation. Since we are in a distributed environment, it is practically impossible to maintain a precise address for every object in every peer. Thus, the ASTs at the peers contain only limited navigation information which may be imprecise. The locating step is repeated on the contacted peers whenever AST is imprecise until the desired peers are reached. The algorithm guarantees that the destination peers are always found. The structure provides a mechanism called image adjustment for updating imprecise parts of AST automatically.

3.2 M-Chord

The Metric Chord (M-Chord) [10] is a decentralized structured P2P network as well but it applies a space transformation rather than a space partitioning. The

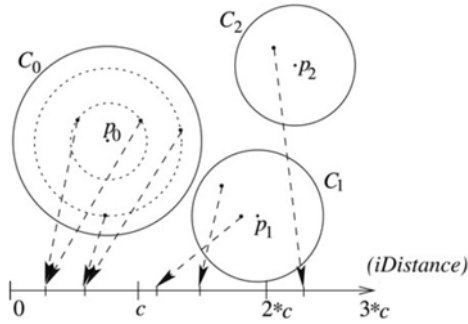


Fig. 5. The mapping principle of M-Chord [10]

transformation maps original objects to numeric identifications that are consequently organized in B^+ -tree. In particular, a set of objects p_0, \dots, p_{n-1} (pivots) are selected and the following transformation based on distances is defined:

$$idistance(o) = d(o, p_i) + i \cdot c.$$

The distance of object o to the closest pivot p_i is determined and along with the separation constant c the numeric address is obtained. Figure 5 visualizes this mapping.

Having the data space mapped into the one-dimensional domain, each peer of the system takes over responsibility for an interval of keys. The structure of the system is formed by the Chord circle [11]. This P2P protocol provides an efficient localization of the peer responsible for a given key. When inserting a new object into the structure, the initiating peer computes the idistance value and employs Chord to forward a store request to the peer responsible for the corresponding interval. The peers store data in B^+ -tree. When a peer reaches its storage capacity limit, it requests a split. A new peer is placed on the Chord circle, so that the requester’s storage is split evenly.

3.3 Scalability Evaluation

In this section, we summarize experience with the approaches described above. We focus mainly on the scalability issue and concurrent query processing. A complete comparison made from other perspectives is available in [9].

Both the structures were implemented as overlays in MUFIN, which allows us to compare them objectively. We used a real-life dataset consisting of 1 million images taken from the CoPhIR dataset [12], for details please refer to Section 5.1. In M-Chord, the transformation using 40 pivots was defined and the capacity of peers’ storage was fixed to 5,000 objects. In case of GHT*, the peers could maintain up to five buckets each of capacity of 1,000 objects. All presented performance characteristics of query processing have been obtained as an average over 100 queries with randomly chosen query objects and the radii of 0.8 (about 100 objects returned).

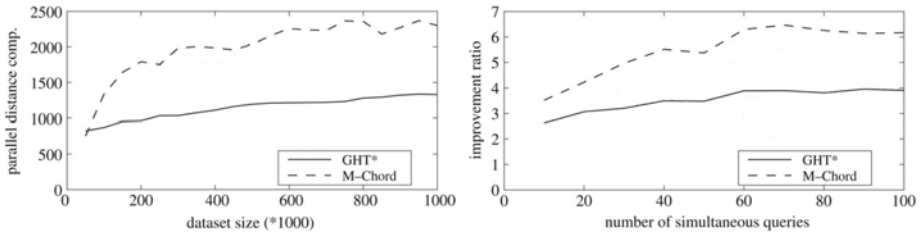


Fig. 6. Increasing data volume: (left) costs in parallel distance computation costs, and (right) query-throughput improvement ratio

Figure 6(left) presents the computational costs in terms of the number of parallel distance computations, i.e. it corresponds to the query response time. The costs grow very slowly. This is caused by the following facts: the peers involved in searching contain more data; and the data space got denser when the volume of data was increased. The noticeable graph fluctuations are caused by quite regular splits of overloaded peers. Figure 6(right) depicts the query-throughput improvement ratio that measures how many queries can be evaluated concurrently without degradation of response time. The differences in the respective improvement ratios are introduced mainly by differences between single-query parallel costs of individual structures. M-Chord handles simultaneous queries noticeably better than GHT*. GHT* employs quite a high number of peers during the query processing, so parallel distance computations are low (see Figure 6(left)). Therefore, simultaneous queries hit the same peers very likely, which increases the overall response time. Furthermore, there is a higher probability in M-Chord that different queries incur load at different peers and, thus, the parallel costs are only marginally increased.

MUFIN inherently supports also centralized index structures. So, the performance of distributed structures can be further improved by organizing peers' local data in a centralized index structure. For example, a very popular solution is to apply M-tree [13] or D-index [14].

4 Unstructured Networks

A technology based on Semantic Overlay Networks (SONs) [15,16,17], which creates a semantic overlay upon an existing unstructured network (e.g. Gnutella), has proven to be useful. The peers sharing similar interests are grouped into semantically similar clusters to improve query performance, while keeping a high degree of peer autonomy. An emerging research direction is to apply principles of self-organizing systems originating from different disciplines such as biology or social sciences. In general, self-organizing systems are characterized by a high degree of scalability, adaptability to changing environment, and robustness to sudden errors. Existing approaches [18,19] applied to search in unstructured networks usually adopt a self-organizing theory of biological systems – the ant-colony system or the social-network theory.

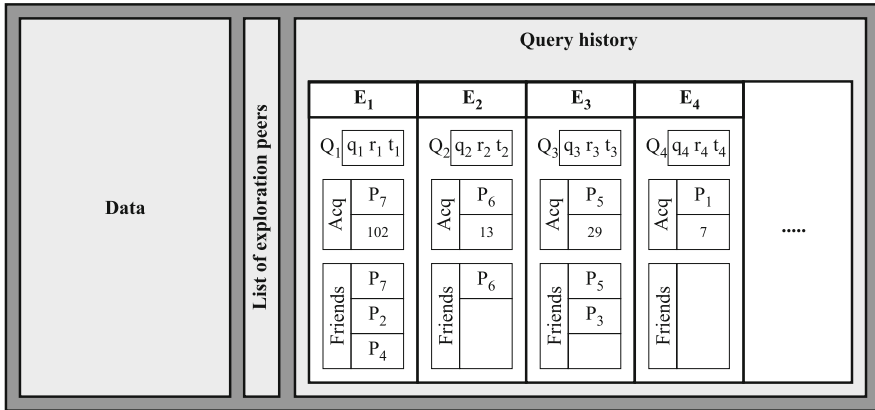


Fig. 7. MSN peer's schema

In this section, we outline a search system called Metric Social Network (MSN) [19]. In particular, MSN can be observed as an overlay implemented as a MUFIN's overlay that operates as an unstructured P2P network. MSN exploits the social-network paradigm [20,21] to lay basics for self-organizing principles – the relationships among peers are established according to analyses of answers of processed queries. An adaptive routing algorithm exploits these relationships for efficient query forwarding. The major difference from the structured-network approaches is that no data distribution principle is imposed, so data need not be transferred to another peer for storage.

Firstly, we summarize the MSN's architecture. Next, we describe its routing algorithm. Finally, we present a sketch of performance evaluation.

4.1 Architecture

Each peer of MSN can organize its own data, can pose similarity queries and must return answers to the queries. Interconnection between peers is based on the query-answer paradigm, i.e., new relationships among peers are established according to answers returned to a processed query. Thus, each peer maintains metadata about queries it has asked or answered, called a *query history*. This represents peer's local knowledge about the network and is exploited by a query-routing algorithm.

A peer P is a tuple (X, H, M) , where X identifies the peer's local database, and $H = \{E_1, \dots, E_n\}$ represents the query history. Individual *entries* E_i identify peers that participated in answering a query Q and form query-specific relationships. In addition, each peer maintains a list of peers M that are employed to explore new and previously unvisited parts of the network. The schema of a peer is depicted in Figure 7.

When a query Q is issued at a peer P_{start} , the routing algorithm tries to locate the most *promising* peers P_1, \dots, P_n in the network. These peers process

the query on their local data and return their answers (*partial answers*) $A_{P_i}(Q)$ to the peer P_{start} . This peer merges the partial answers and returns the *combined answer* to the user, denoted as $A(Q) = \bigcup_{i=1}^n A_{P_i}(Q)$. Remark that the combined answer is approximate. To determine which peer answered better, the quality of the partial answers has to be measured. Even though sophisticated quality measures can be defined, MSN uses the quality of peer's answer expressed simply as the number of retrieved objects, i.e. $|A_{P_i}(Q)|$.

Two kinds of relationships are distinguished. Firstly, the acquaintanceship denotes that the target of the relationship is the best peer (*acquaintance*) to answer the given query. The acquaintance has the highest quality of the answer to the query Q and is defined as follows:

$$Acq(Q) = P \Leftrightarrow \forall P_i : |A_P(Q)| \geq |A_{P_i}(Q)|,$$

for $i \in \{1, \dots, n\}$ where n denotes the number of peers answering the query Q . Secondly, the friendship represents the similarity of peers – two peers are *friends* when they give a similar (high-quality) answer to the query Q .

$$Fri(Q) = \{P_i : |A_{P_i}(Q)| \geq |A(Q)|/n\}.$$

Note that the acquaintance and the best friend are the identical peer.

After processing the query Q , each peer P_i identified as a friend stores a new entry E in its query history. This entry $E = (Q, Acq(Q), |A_{Acq(Q)}(Q)|, Fri(Q))$ is a tuple, where $Q = R(q, r, t)$ denotes the range query with timestamp, $Acq(Q)$ is the acquaintance, $|A_{Acq(Q)}(Q)|$ is its quality, and $Fri(Q)$ is the set of friends. The query-issuing peer P_{start} and peers contacted as exploration peers store this entry as well, but the set of friends is empty unless the particular peer has also been identified as a friend.

4.2 Adaptive Query Routing

In this part, we describe an adaptive query-routing algorithm proposed in [22] that enables each peer to control routing according to its current knowledge. In principle, each peer that is asked to process a query checks its query history for the most relevant entries. Next, the peer forwards the query to the acquaintances of these entries or evaluates the query on the local data and contacts friends. If there are few relevant entries only or there are not any, the routing algorithm uses exploration peers to locate unvisited peers that may contain the required data.

Relevancy of Entries. The relevancy of entries is measured by *confusability* of two queries – the query being evaluated and a query stored within an entry in the query history. The confusability function is a continuous function and returns a real value within $[0, 1]$. The higher the value is returned, the more confusable (relevant) the queries are. If it returns 1, the queries are identical. The function takes into account the distance between query objects of queries, their query radii and the time when the queries were issued. The time aspect is important to allow aging information about the peer's neighborhood. The formal definition called adaptive gaussian-like confusability is available in [22].

Exploration. The design of MSN incorporates factors to improve quality of query answers and to allow new peers to join the system efficiently. Each peer of MSN maintains its list of exploration peers over time [23]. At the beginning, it has to know at least one existing peer in order to be able to forward a query to other peers. The routing algorithm exploits this list in a way that it contacts not only the most promising peers retrieved from the query history but also some exploration peers that help find new and unvisited parts of the network.

Routing Algorithm. In general, a new query is being forwarded to the peers that should have better *knowledge* about the query – knowing more-promising peers or containing relevant data. The peer’s knowledge is interpreted as confusability (P^{conf}) and for query-issuing peer P_{start} is set to zero ($P_{start}^{conf} = 0$). Firstly, P_{start} goes through its query history, computes the values of confusability between a new query $Q = R(q, r, t)$ and queries of all stored entries, and returns the entries descendingly ordered by confusability. Secondly, the list of relevant entries E_{rel} is constructed. All entries having confusability ≥ 0.8 are added to E_{rel} , because they are highly relevant to Q . If there are fewer entries in E_{rel} than 5, next entries having confusability ≥ 0.3 are added to fill up E_{rel} to contain five entries. Next, each entry in E_{rel} is processed as follows:

- If the entry has confusability C higher than the current peer’s confusability P^{conf} , the query is forwarded to the acquaintance P_{acq} picked from this entry and its confusability P_{acq}^{conf} is set to C .
- Otherwise the query is not forwarded and is processed on local data. In addition, friends of entries in E_{rel} that have confusability ≥ 0.8 , are asked to evaluate Q on their local data too. It is supposed that these friends hold substantial parts of the total answer $A(Q)$. The partial answers are finally returned to P_{start} .

If the list E_{rel} is shorter than five entries or even does not contain any entry, Q is forwarded to up to five exploration peers. To avoid flooding the network, forwarding to exploration peers is stopped after a predefined number of hops is reached (in our case, 3 hops). The complete specification of adaptive query routing algorithm is available in [22].

4.3 Adaptability and Robustness Evaluation

In order to study characteristics of the query routing algorithm, we have implemented MSN in MUFIN and executed real-life experiments. We used 100,000 images taken from the CoPhIR dataset [12], described in details in Section 5.1. Each of image has its owner ID associated, so we distributed images over P2P network in a way that each peer contains images of one Flickr user. Because there are high differences in the number of images taken by individual users, we have split overfilled peers. As a result, we obtained 2,000 peers each organizing 50 images of the same Flickr user.

Figure 8(left) reports on the results obtained by repeating 100 times a batch of queries and measuring performance indicators. In particular, we measured

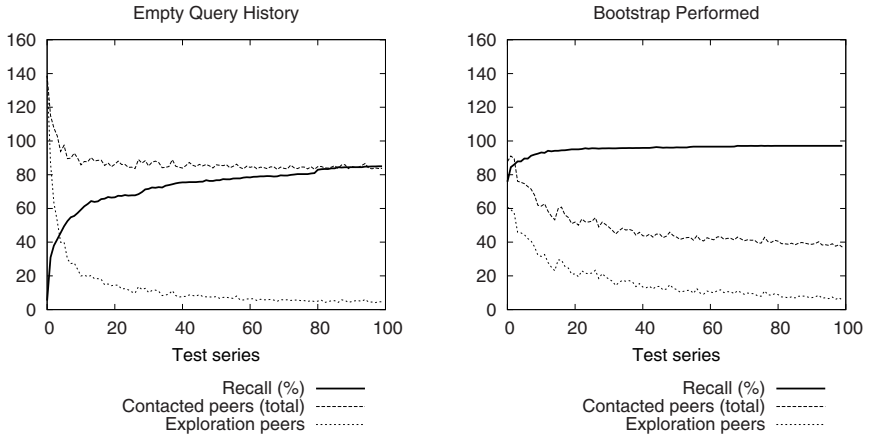


Fig. 8. Performance indicators of MSN: (left) query history is initially empty, and (right) query history is populated with 3 random queries

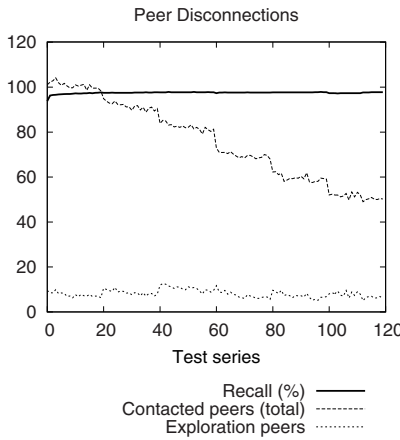


Fig. 9. Performance indicators of MSN: gradual disconnection of peers

recall, the number of exploration peers used during querying, and costs in terms of contacted peers (peers participated in query evaluation including exploration peers). The batch consisted of 50 range queries with randomly picked query objects and varying radii. From the figure, we can read that the system starting from zero knowledge (peers had query histories empty) started to evolve and the recall has reached 85% while contacting less than 90 peers. Initially, each peer had just 50 exploration peers, so the peers could use only exploration peers for query routing.

Performance of MSN can radically change if a peer joining the system proceeds a bootstrap procedure. Figure 8(right) shows the same experiment but the peers

performed the following bootstrap procedure during their joining. Firstly, three objects were picked at random from the peer's local data. Secondly, range queries with these objects and radius 0.8 were posed. Finally, the MSN evaluated the queries. This helps distribute the knowledge about new peer's local data in the network. As a result, the first batch execution reached 80% recall. After 100th batch execution, the recall was 97% and the costs decreased to 37 contacted peers. In this way, the quality of service of MSN is greatly improved.

We have also tested robustness of MSN by gradually disconnecting up to 1,000 peers. Figure 9 shows the same performance indicators when 200 random peers got disconnected forcibly after each 20th batch execution. The most interesting fact about the recall curve is that it stays almost constant. This proves adaptability of the query routing algorithm and robustness of the whole system. The answer was degrading in terms of amount of retrieved data, but only because some data became unavailable.

5 Prototype Applications

As mentioned in the previous sections, the similarity search approach used in MUFIN is highly universal and extensible. In this section, we describe several application domains where MUFIN can be used. However, due to MUFIN's versatility this list is not complete. Firstly, we present a large-scale image retrieval demo. Next, we summarize other applications and give ideas how to incorporate them into MUFIN.

5.1 Large-Scale Image Search

This application [24] represents a possible instance of MUFIN for content-based similarity search in a large collection of general images available on the internet. In particular, the dataset consists of 100 million images taken from CoPhIR Database [12]. Each image is represented by five global MPEG-7 descriptors [25], namely *color structure* (CS), *color layout* (CL), *scalable color* (SC), *edge histogram* (EH), and *homogeneous texture* (HT). Specifically, CS, CL, and SC express the spatial distribution of colors in an image. The EH captures local density of edge elements and their directions (sometimes called the *structure* or *layout*); it acts as a simple and robust representation of shapes. Finally, HT is a texture descriptor. These descriptors are represented as vectors and the MPEG-7 standard defined a specific distance measure for each of them. These measures satisfy the metric postulates and they are aggregated into a single distance function. The whole dataset is organized in M-Chord and peers' local data are stored in M-tree. For details, please refer to Section 3.2. An example of retrieving k images which are the most similar to a given query image is given in Figure 10. For further details, please refer to [27].

5.2 Biometric Applications

In general, biometrics are automated methods of recognizing a person based on the person's physiological and behavioral characteristics. Biometrics include a

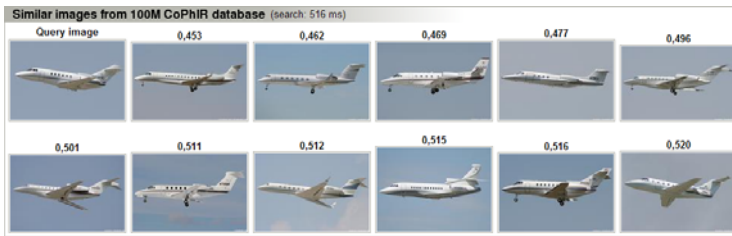


Fig. 10. Image Retrieval: the result of a query



Fig. 11. Example of minutiae extracted from a fingerprint image [26]

wide variety of technologies ranging from traditional fingerprints over facial or iris recognition and retinal scanning to DNA testing, speech verification and gait recognition. MUFIN can be applied to the problem of *identification*, the aim of which is to tell who the person that exposes its biometric characteristic is.

A famous application of biometrics is in criminalistics and in border and immigration control where fingerprints are compared. Minutiae is one of the successfully applied methods of comparing ridges in fingerprints [26]. It identifies places where ridges start, stop or bifurcate (branch), refer to Figure [11]. These places are then observed as points with a direction and are converted to polar coordinates. As a result, a fingerprint is described as a sequence of points. Two sequences are then matched using a weighted edit distance function. The used weights do not break metric postulates, so this distance function is directly applicable to MUFIN.

Gait, or the way a person walks, is a unique and idiosyncratic characteristic of the person. Its advantage for biometrics is that it is difficult to conceal and it can be easily captured even at long distances. In [27], the gait information is extracted from a video sequence. In particular, a silhouette of the walking person is determined for each video frame by subtracting the background of the image. The sequence of silhouettes is divided in subsequences each of them representing one gait cycle (two steps). Then, an average silhouette is computed for each subsequence, see Figure [12]. The binary silhouettes are then compared using the Euclidean distance, which is metric.

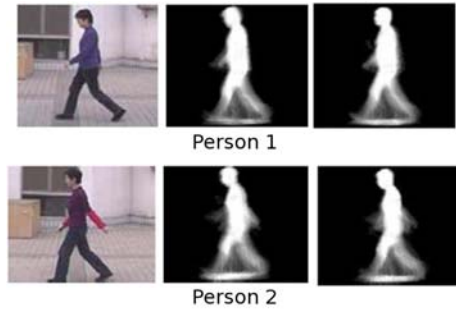


Fig. 12. Example of average silhouette extraction [27]

6 Conclusions

There are no doubts that modern similarity search in computer networks needs new technology to apply. In this paper, we have shortly introduced MUFIN, an approach to similarity searching, which is designed on concepts of: (i) extensibility - to achieve applicability to different collections comparing data by various measures of similarity; (ii) scalability - to process extremely large collections of data queried by many concurrent requests; (iii) infrastructure independence - to tune performance according to needs of specific applications. The implementation on structured P2P networks is able to achieve quality of service by tuning the performance according to specific application needs. We discuss several structured P2P protocols, all of them running with logarithmically bound number of hops. Local data on peers is organized in centralized metric similarity search structures. Unstructured P2P networks with high degree of peer churning are considered as systems of self-organizing peers for which a social network of search requests and answers is built. Such architecture can learn and improve its effectiveness in time; it is also able to react to the changing number of peers properly. Important features are demonstrated by an on-line demo available from <http://mufin.fi.muni.cz/imgsearch/>.

Acknowledgments. This research was partially supported by the Czech Science Foundation projects 201/09/0683 and 201/07/P240. The access to the MetaCentrum (super)computing facilities provided under the research intent MSM6383917201 is also appreciated.

References

1. Novak, D., Batko, M., Zezula, P.: Generic similarity search engine demonstrated by an image retrieval application. In: The 32nd Annual International ACM Conference on Research and Development in Information Retrieval, p. 840. ACM Press, New York (2009)

2. Batko, M., Dohnal, V., Novak, D., Sedmidubsky, J.: MUFIN: A Multi-Feature Indexing Network. In: The 2nd International Workshop on Similarity Search and Applications, pp. 158–159. IEEE Computer Society, Los Alamitos (2009)
3. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. In: *Advances in Database Systems*, vol. 32. Springer, Heidelberg (2006)
4. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: *DELOS Conference 2007: Working Notes*, pp. 11–23. Information Society Technologies (2007)
5. Batko, M., Kohoutková, P., Zezula, P.: Combining metric features in large collections. In: *The 1st International Workshop on Similarity Search and Applications*, pp. 79–86. IEEE Computer Society, Los Alamitos (2008)
6. Amato, G., Rabitti, F., Savino, P., Zezula, P.: Region proximity in metric spaces and its use for approximate similarity search. *ACM Transactions on Information Systems* 21(2), 192–227 (2003)
7. Novak, D., Batko, M., Zezula, P.: Web-scale system for image similarity search: When the dreams are coming true. In: *The 6th International Workshop on Content-Based Multimedia Indexing*, pp. 446–453. IEEE, Los Alamitos (2008)
8. Litwin, W., Neimat, M.A., Schneider, D.A.: LH* – a scalable, distributed data structure. *ACM TODS* 21(4), 480–525 (1996)
9. Batko, M., Novak, D., Falchi, F., Zezula, P.: Scalability comparison of peer-to-peer similarity search structures. *Future Generation Computer Systems* 24(8), 834–848 (2008)
10. Novak, D., Zezula, P.: M-Chord: A scalable distributed similarity search structure. In: *The 1st International Conference on Scalable Information Systems*, pp. 1–10. IEEE Computer Society, Los Alamitos (2006)
11. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: *The 2001 ACM Conference on Applications, Technologies, Architectures, Protocols for Computer Communications*, pp. 149–160. ACM Press, New York (2001)
12. Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F.: CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2 (2009)
13. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: *The 23rd International Conference on Very Large Data Bases*, pp. 426–435. Morgan Kaufmann, San Francisco (1997)
14. Dohnal, V., Gennaro, C., Savino, P., Zezula, P.: D-Index: Distance searching index for metric data sets. *Multimedia Tools and Applications* 21(1), 9–33 (2003)
15. Aberer, K., Cudré-Mauroux, P.: Semantic overlay networks. In: *The 31st International Conference on Very Large Data Bases*, p. 1367. ACM Press, New York (2005)
16. Bender, M., Crecelius, T., Kacimi, M., Michel, S., Parreira, J.X., Weikum, G.: Peer-to-peer information search: Semantic, social, or spiritual? *IEEE Data Eng. Bull.* 30(2), 51–60 (2007)
17. Crespo, A., Garcia-Molina, H.: Semantic overlay networks for p2p systems. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) *AP2PC 2004*. LNCS (LNAI), vol. 3601, pp. 1–13. Springer, Heidelberg (2005)
18. Michlmayr, E.: Self-organization for search in peer-to-peer networks: the exploitation-exploration dilemma. In: *The 1st international conference on Bio inspired models of network, information and computing systems*, p. 29. ACM Press, New York (2006)

19. Sedmidubsky, J., Bartoň, S., Dohnal, V., Zezula, P.: A self-organized system for content-based search in multimedia. In: The IEEE International Symposium on Multimedia, pp. 322–327. IEEE Computer Society, Los Alamitos (2008)
20. Wasserman, S., Faust, K., Iacobucci, D.: Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press, Cambridge (1994)
21. Granovetter, M.: The strength of weak ties. *American Journal of Sociology* 78(6), 1360–1380 (1973)
22. Dohnal, V., Sedmidubsky, J.: Query routing mechanisms in self-organizing search systems. In: The 2nd International Workshop on Similarity Search and Applications, pp. 132–139. IEEE Computer Society, Los Alamitos (2009)
23. Sedmidubsky, J., Bartoň, S., Dohnal, V., Zezula, P.: Querying similarity in metric social networks. In: Enokido, T., Barolli, L., Takizawa, M. (eds.) NBIS 2007. LNCS, vol. 4658, pp. 278–287. Springer, Heidelberg (2007)
24. Batko, M., Falchi, F., Lucchese, C., Novak, D., Perego, R., Rabitti, F., Sedmidubsky, J., Zezula, P.: Building a Web-scale Image Similarity Search System. *Multimedia Tools and Applications*, 31 (2009)
25. Manjunath, B.S., Salembier, P., Sikora, T. (eds.): Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons, Inc., New York (2002)
26. Jain, A.K., Maltoni, D.: Handbook of Fingerprint Recognition. Springer-Verlag New York, Inc., Secaucus (2003)
27. Fazenda, J., Santos, D., Correia, P.: Using gait to recognize people. In: The International Conference on Computer as a Tool, vol. 1, pp. 155–158. IEEE Press, Los Alamitos (2005)

Network Attack Detection Based on Peer-to-Peer Clustering of SNMP Data*

Walter Cerroni, Gabriele Monti, Gianluca Moro, and Marco Ramilli

DEIS – University of Bologna, v. Venezia 52, 47521 Cesena (FC), Italy
{walter.cerroni,gabriele.monti4,gianluca.moro,marco.ramilli}@unibo.it

Abstract. Network intrusion detection is a key security issue that can be tackled by means of different approaches. This paper describes a novel methodology for network attack detection based on the use of data mining techniques to process traffic information collected by a monitoring station from a set of hosts using the Simple Network Management Protocol (SNMP). The proposed approach, adopting unsupervised clustering techniques, allows to effectively distinguish normal traffic behavior from malicious network activity and to determine with very good accuracy what kind of attack is being perpetrated. Several monitoring stations are then interconnected according to any peer-to-peer network in order to share the knowledge base acquired with the proposed methodology, thus increasing the detection capabilities. An experimental test-bed has been implemented, which reproduces the case of a real web server under several attack techniques. Results of the experiments show the effectiveness of the proposed solution, with no detection failures of true attacks and very low false-positive rates (i.e. false alarms).

Keywords: Network security, distributed intrusion detection, SNMP, data mining, data clustering, peer-to-peer.

1 Introduction

Network security is one of today's most important issues that must be dealt with by system engineers in their everyday work as well as by the research community. In particular, the problem of detecting malicious network traffic and promptly trigger alerts and/or suitable countermeasures has been widely studied in the last decade and is still of high interest. To this purpose, Network-based Intrusion Detection Systems (NIDSs) [7] have been developed with the ability to analyze network traffic, detect possible attacks and notify the network administrators. The NIDS operations are executed according to two possible approaches, respectively signature-based and anomaly-based [15].

The first approach relies on the idea that, by comparing well-known malicious network behaviors with the current network activity by means of traffic signatures, it is possible to detect the presence of harmful traffic with a good

* Work partially funded by the european project DORII: Deployment of Remote Instrumentation Infrastructure Grant agreement no. 213110.

level of confidence and reliability. Unfortunately, signature-based schemes suffer from the so-called synonymous attack, where the attacker is able to bypass the signature check by using a different stream pattern with the same harmful meaning.

Anomaly-based NIDSs, on the other hand, are capable of detecting a threat by looking at the specific behavior of the network traffic: what is known is considered as “normal” activity, whereas any behavior that differs from normal traffic is considered as anomaly. The most difficult challenge of these systems is to figure out what is actually normal activity and what is not. In particular, this approach becomes tricky and very difficult to apply to networks characterized by heterogeneous user behaviors and highly variable traffic patterns.

A significant research effort has been spent in the last few years with the objective of increasing NIDS efficiency. For instance, by applying fuzzy logic to intrusion detection [8] [3] or by adopting an approach based on artificial neural network [30] [16]. Other solutions include the use of an agent-oriented paradigm to build a multi-agent system able to detect threats [31] or the development of an embedded NIDS inside a Network Interface Card (NIC) [11]. Finally other studies focused their attention on software engineering aspects of intrusion detection [26].

A common assumption made in most of the published work on NIDS is the analysis of network traffic through raw packet capture techniques. However, this is a very critical aspect, since packet-by-packet analysis may become a system bottleneck in case of very large traffic volumes. In fact, some packet sampling techniques have been recently investigated [19] [1] that are seeking a trade-off between detection accuracy and performance. In some cases, using raw packets it is not even possible to distinguish normal traffic from Denial of Service (DoS) attacks [22].

A viable alternative to raw traffic capture performed by NIDSs is the use of the monitoring facilities provided by the Simple Network Management Protocol (SNMP), the de-facto standard adopted in Network Management Systems (NMS) [12]. A first proposal for a methodology that integrates NMS and NIDS has been introduced with reference to proactive detection of Distributed Denial of Service (DDoS) attacks [4]. Other studies include anomaly detection using signal processing techniques on SNMP data [25] and SNMP-based traffic flooding attack detection [29].

The contribution of this paper is to follow a new approach based on data mining techniques, in particular applying data clustering to information collected through SNMP. In this context, one of the data clustering peculiarities is the capacity to perform successful detection without a training phase, which, instead, is required by all the supervised techniques, such as the popular ones based on decision trees. The training phase is a costly and time-consuming activity because a significant amount of data must be correctly classified in advance by human experts.

The experiments performed on a test-bed using real traffic traces show that the proposed methodology is capable of detecting many different network attacks,

such as DoS, DDoS and several flavors of TCP port scanning, with a very high accuracy, with no detection failures of true attacks and very low false alarm rates. In addition, a thorough analysis driven by the adopted data mining approach allows to understand which pieces of information collected through SNMP are really essential for attack detection.

The paper is organized as follows. Section 2 provides a brief overview of data clustering techniques, with particular reference to the k-means approach used in this paper. Then section 3 describes the architecture and operations of the proposed framework, followed by section 4 which presents the experimental test-bed and the obtained results. Finally, section 5 concludes the work.

2 Background on Data Clustering

Data mining is the extraction of implicit, previously unknown, and potentially useful information from data [10]. Data mining techniques are usually divided in unsupervised and supervised according to the learning (i.e. the information extraction) method adopted. The unsupervised mining, differently from the supervised one, do not require training phases saving the classification cost performed by human experts to define a valid training set.

The goal of data clustering, which is the unsupervised technique included in our framework, is to extract new potential useful knowledge from a generally large data set by grouping together similar data items and by separating dissimilar ones according to some defined dissimilarity measure among the data items themselves. The literature on data clustering offers a large number of algorithms, generally grouped in hierarchic (e.g. BIRCH [32]), density-based approaches (e.g. DENCLUE [13]), linkage-based, statistics-based methods and partitive solutions (e.g. k-means [18]).

The hierarchic methods can be further divided in agglomerative (i.e. bottom-up) or divisive (i.e. top-down), according to how the algorithms begin the formation of groups, namely with each element as a separate cluster which is gradually merged into successively larger clusters, or alternatively dividing the whole set into successively smaller clusters.

In density-based approaches the idea is that similarity is expected to be high in densely populated regions of the given data set. Consequently, searching for clusters may be reduced to searching for dense regions of the data space separated by regions of relatively lower density. Popular methods in this class have been investigated in the context of non-parametric density estimation [24] and data mining [9, 13, 28].

Partitive approaches, in particular k-means that has been used in the proposed framework test-bed, aim to partition observations into k clusters specifying randomly in advance k centroids (cluster centers). Each observation is associated to its closest centroid according to a distance metric and then each centroid updates its position according to its associated observations; the process iterates until the k centroids no longer change their positions. Once the iteration stops, each point is assigned to its nearest cluster center, so the overall effect is to minimize the

total squared distance from all points to their cluster centers. In general this is a local minimum and the final result depends on the initial position of k centroids, however there are valid heuristics to select their positions to achieve suboptimal solutions [2]. In general it is almost infeasible to find globally optimal clusters with any kind of clustering algorithms.

A description of each group of solutions above mentioned, which is beyond the scope of this paper, is available in [27].

The data clustering problem has been investigated also in the distributed setting where data cannot be concentrated on a single machine, for instance because of privacy reasons or due to network bandwidth limitations, or because of the huge amount of distributed data. Several algorithms have been developed for distributed data clustering [14] [17] [23]. A common scheme underlying all approaches is to first locally extract suitable aggregates, then send the aggregates to a central site where they are processed and combined into a global approximate model. The kind of aggregates and combination algorithm depend on the data types and distributed environment under consideration, e.g. homogeneous or heterogeneous data, numeric or categorical data.

A k-means algorithm for clustering data distributed over a large, dynamic network is presented in [6], suited for overlay peer-to-peer systems [21] [20]. The algorithm requires only local communication and synchronization at each iteration, namely each node cooperates only with its neighboring nodes. Authors achieved high accuracy levels with less than 3% on average of misclassified with respect to the centralized version of k-means.

3 Description of the Proposed Framework

The reference scenario considered in the attack detection approach proposed in this paper is sketched in Fig. 1. The basic idea is to have several monitoring stations to share their knowledge of the traffic behaviors and their attack detection capabilities according to any peer-to-peer (P2P) collaborative paradigm; namely according to any unstructured or structured P2P overlay network. Each monitor is based on a standard SNMP management station configured to collect traffic data from a number of SNMP agents running on hosts, servers, workstations, laptops, etc. This is a very common situation, as most organizations are using SNMP to manage their networks.

Data collected from SNMP agents are represented as objects according to a standard language (ASN.1) and organized in a tree-structured database called Management Information Base (MIB). Each MIB object provides information about the corresponding feature being managed, e.g. the number of packets received on a network interface, the amount of disk space available on a server, the availability of a given service and so on. In particular, for the purpose of the methodology presented here, the MIB objects related to IP and TCP are considered.

Besides the typical network management tasks that may or may not be implemented, each monitoring station uses the queried MIB objects to extract its

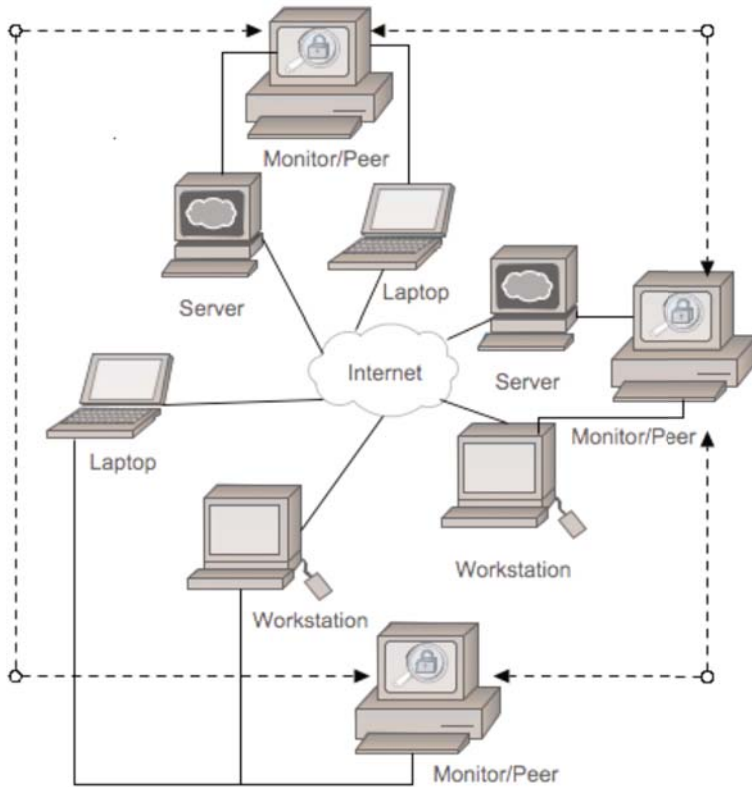


Fig. 1. Reference scenario of the SNMP-based attack detection framework

knowledge of the network behavior, according to which it is able to distinguish normal traffic and several kinds of attack. This process is performed by applying a clustering algorithm to observations whose schema (i.e. the relevant variables) and the corresponding instances are derived from collected SNMP data. The knowledge (i.e. the data clustering model) is represented as a set of centroids (i.e. cluster centers), therefore the memory required is less than a couple of Kilo-bytes and this guarantees high detection efficiency once the clustering model is applied to new incoming SNMP data. In other words, the framework can work in real-time manner.

Each monitor then periodically collects the content of the network-related MIB objects and process them by updating the clustering model in background. The model updates improve the detection accuracy. In fact an increasing effectiveness has been observed in the test-bed described in the next section when the SNMP data set becomes larger.

Figure 2 shows the methodology adopted by the implemented framework working as a monitor. The software running on the monitor machine reads the SNMP TCP stack information from the monitored machines and generates the

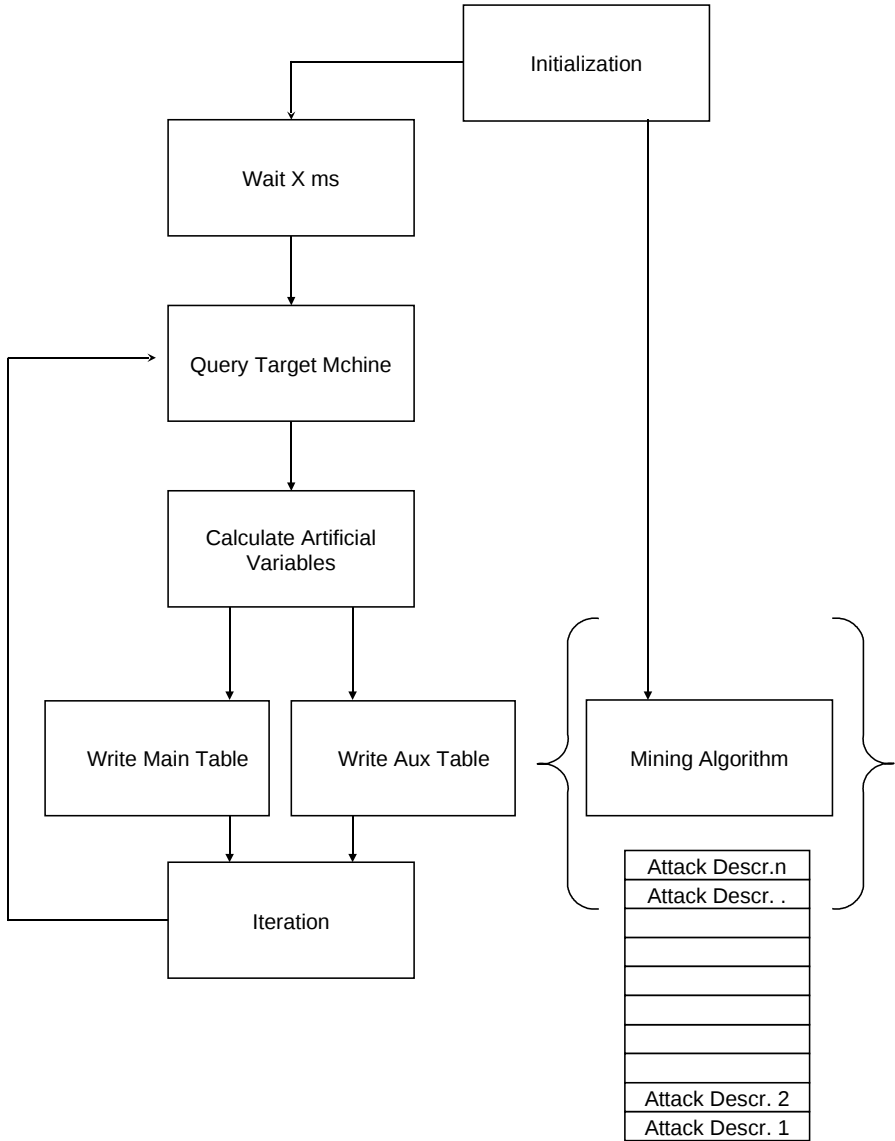


Fig. 2. Logical software flow

observations producing two tables used by the data clustering algorithm. One table, called the *main table*, includes general TCP stack information as well as some computed variables, both useful to discern attacks from normal traffic. The other table, called the *auxiliary table*, summarizes some correlations from the main table related to connected hosts and connection details, useful to differentiate attacks.

Table 1 reports the list of relevant variables of each observation that have proved to guarantee high and stable levels of detection accuracy; we have defined them by evaluating combinations of hundreds of SNMP parameters by using several mining techniques for feature extractions. This scheme of variables determines a multi-dimensional space, where each variable represents a dimension and each observation is a point whose coordinates correspond to the variable values.

Table 1. Relevant Clustering Variables Derived from SNMP Data

<i>Features Derived from SNMP Data</i>
Number of processes in TCP listen state
Number of open TCP connections (any possible TCP state)
Number of TCP connections in time-wait state
Number of TCP connections in established state
Number of TCP connections in SYN-received state
Number of TCP connections in FIN-wait state
Number of different remote IP addresses with an open TCP connection
Remote IP address with the highest number of TCP connections
Remote IP address with the second highest number of TCP connections
Remote IP address with the third number of TCP connections
Local TCP port with the highest number of connections
Number of connections to the preceding TCP port
Local TCP port with the second highest number of connections
Number of TCP RST segments sent out

More specifically, the clustering algorithm we used in the test-bed of this framework is the k-means, introduced in the previous section. The number of clusters specified in advance must be two or more in order to learn a model able to at least discern normal traffic from attacks. In general, the number of clusters should correspond to the number of different attacks to be detected plus one. However, it is important to clarify that the clustering model does not indicate which cluster corresponds to which attack. This meaning association occurs by interpreting the knowledge discovered. Anyway, the same kind of attacks perpetrated against different machines using the same features, like those we have introduced above, become points which fall in the same zone of the multi-dimensional space, leading naturally to similar clusters everywhere in the P2P network.

With our framework this convergence of clusters in the P2P network is further enhanced thanks to the collaboration among peers. In fact, each peer, i.e. each monitoring station, may share with one or more neighbours its observations, which do not represent any network transmission content, or it may simply share its knowledge, namely its local cluster centroids with its cardinality (i.e. the number of associated observations). In the latter case, the traffic among peers is almost negligible since it corresponds to less than a couple of Kilobytes.

Moreover, the frequency of this information exchange is as low as the number of times the local knowledge is updated, therefore even the sharing of observations is a practicable method. The observations coming from one or more neighbours are simply added to the local ones in order to contribute to improve the next update of the local clustering model. The same happens with the transmission of cluster centroids. In the first case the clustering algorithm behaves as usual updating its centroids using the new observations together with its local ones, while in the second case its local centroids are updated according to the weight (i.e. the cardinality) of the received centroids as well.

4 Test-Bed Setup and Results

To prove the feasibility and accuracy of the proposed network attack detection methodology, an experimental test-bed has been set up emulating a typical situation where some standard web servers might be under attack. Fig. 3 shows the particular scenario where a machine controls what is happening on the monitored server. This scenario has been reproduced ten times to collect collaborative data of ten distinct servers for a real consistent experiment. The web server is connected to a De-Militarized Zone (DMZ), whereas legitimate clients as well as attacking hosts from the external network are able to contact the server through a router. The monitoring station is connected to the server through a separate private network, which is also a typical network management situation where the monitoring and management traffic is kept safe and isolated from the public Internet, e.g. on a dedicated VLAN.

Another machine has been used to generate synthetic traffic replicating requests directed to a web server according to real traces collected on a public

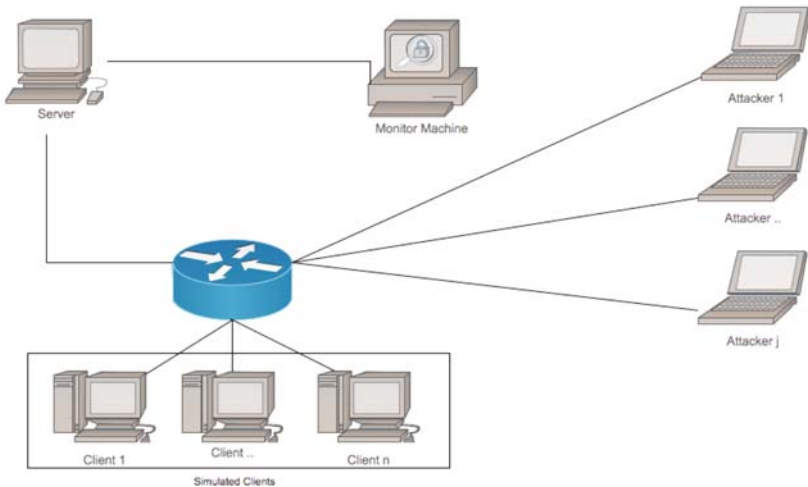


Fig. 3. Test-bed setup: One cell to which corresponds one peer

backbone link [5]. With this approach it was possible to emulate the timings of a realistic and significant amount of HTTP traffic under a controlled environment. A bunch of additional machines has then been set up to perform several different kinds of network attacks:

1. Denial of Service
2. Distributed Denial of Service
3. TCP Port Scanning using different techniques: FIN, SYN, ACK, WINDOW, NULL, XMAS
4. SSH Denial of Service
5. SSH Brute Force

The experiment has been executed in five different sessions, plus a session of normal traffic only. For each session, 1000 samples of the network-related MIBs have been collected and stored in the main and auxiliary tables on the local file system. The tables have been further processed to include, besides the natural SNMP MIBs, some more specific information according to Table 1. All these variables are useful to figure out which host might attack the monitored system. Once the monitor has collected enough data, it is ready to communicate its results to other monitor peers. The communication can be performed by sending all collected data or the learned models only, as previously explained.

The following section describes the results of all possible scenarios emulated through different kinds of simulations, such as: a non collaborative host, two collaborative hosts, three collaborative hosts and so forth until nine collaborative hosts for each peer.

4.1 Results

To validate the results, in order to measure the efficacy of our framework, the observations generated from SNMP data, have been labeled according to the belonging network attack session, including the one of normal traffic, as mentioned in the previous section.

We highlight that the observation labeling has been totally ignored by the data clustering algorithm during the model learning, just because the approach is not supervised. The labels have been used only in test phases to compute the efficacy of the clustering model in the following two cases:

1. for discerning attacks from normal traffic, without distinguishing the kind of attack;
2. for detecting even the kind of attack together with the normal traffic.

Formally the accuracy is defined as follows:

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP is the number of true positive observations, namely the number of attacks correctly detected as attacks, TN is the number of observations correctly

detected as normal traffic, FP regards the false positives, that is the amount of normal traffic erroneously detected as attacks and finally the false negative, namely attacks wrongly interpreted as normal traffic.

Moreover the following rates represent the incidence of false alarms and of undetected alarms (i.e. detection failures of true attacks):

$$\frac{FP}{TN + FN} \tag{2}$$

$$\frac{FN}{TP + FP} \tag{3}$$

$$\frac{FP}{TP + FP} \tag{4}$$

Usually the variables of expression (2)-(4) are represented in a squared matrix, called *confusion matrix*, in which the numerators are along a diagonal, moreover FP and FN are in the same row with TP and TN respectively. In the two kinds of test phases above mentioned, we have computed a series of both 2x2 and 6x6 confusion matrixes respectively.

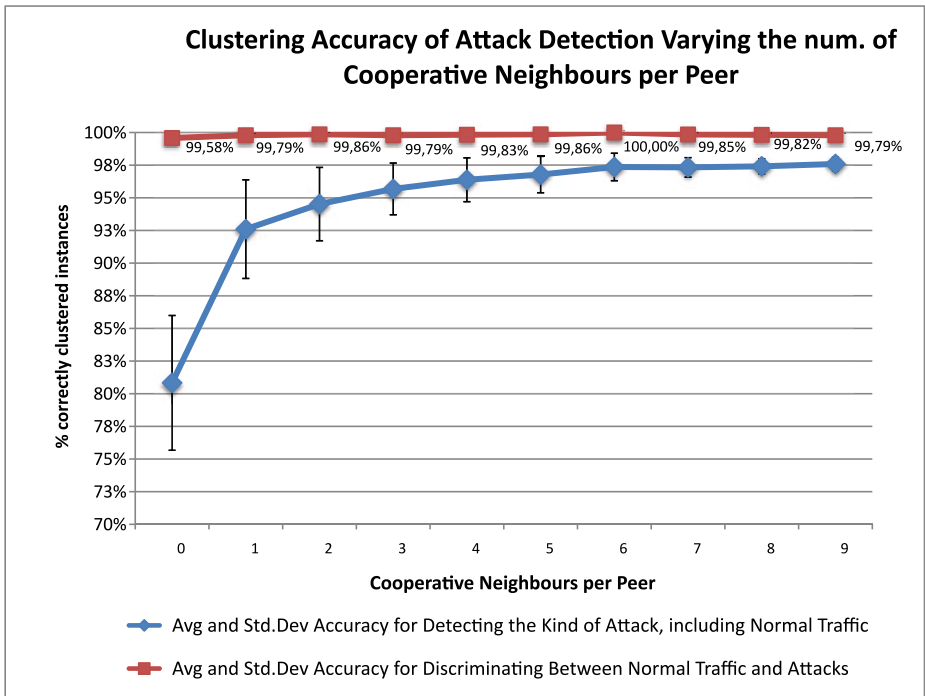


Fig. 4. Detection Accuracy Based on SNMP Data Clustering Varying the num. of Cooperative Neighbours per Peer

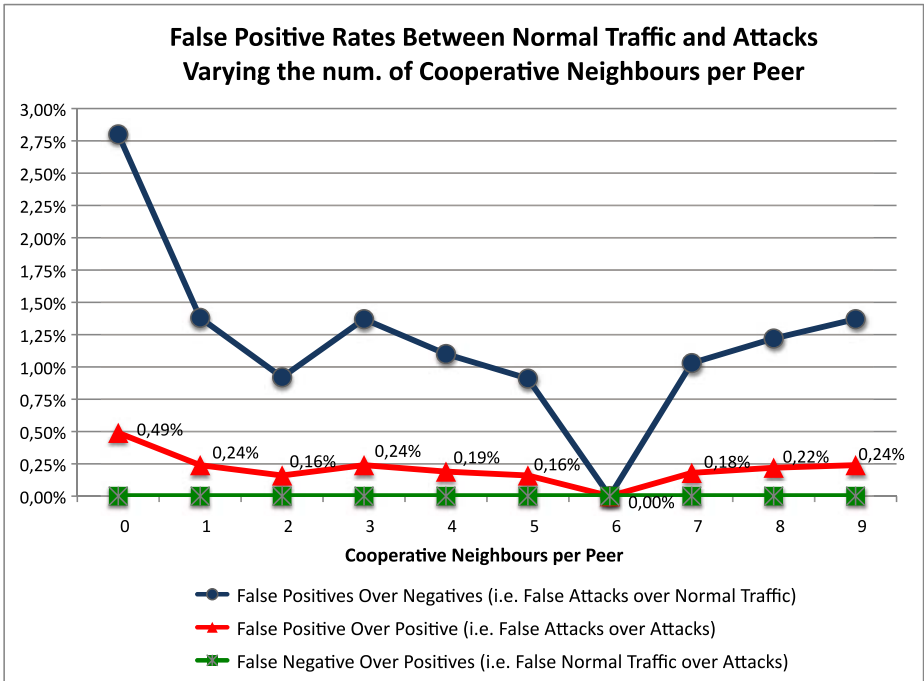


Fig. 5. False Positive Rates Between Normal Traffic and Attacks To Measure False Alarms Varying the num. of Cooperative Neighbours per Peer

Experiments and corresponding measures have been performed by varying, for each peer, the number of collaborative neighbours from zero (i.e. no collaboration) to nine, moreover the same experiment has been repeated ten times with different random seed in order to compute average values and standard deviations.

Figure 4 shows the two series of accuracy corresponding to the two test phases above mentioned. In both series the greatest accuracy increase occurs from zero to one collaborative neighbour. The accuracy in the experiments regarding the discerning between normal traffic and attacks is in the worst case 99.58%, while in the best case is 100%. In the experiments for the detection also of the kind of attack, the worst accuracy is 80.8% without any collaboration, while the best one is 97.6% with nine collaborative neighbours. Another interesting result is that the standard deviations of both series decrease by increasing the number of collaborative neighbours.

Figure 5 illustrates the rates about false alarms and undetected attacks according to the expressions 2, 3 and 4 varying the number of cooperative neighbours per peer. The first important results is that, according to expression 4, the rate of false negative over positives, namely the undetected attacks, is always zero. Moreover the false allarms, corresponding to normal traffic erroneously

detected as attacks, decreases drastically from 2.80%, which is the worst result, to 1.38% with only one cooperative neighbour; this rate on average is 1.21% and its best value is 0%. The incidence of false attacks over attacks is always less than 0.5%.

Finally, in the test-bed, we have observed that the accuracy of clustering models is very well preserved over new incoming observations. In fact, the loss of accuracy, in case of missing model updates, is on average only 0.39%, when the amount of new observations, generated from new network traffic, is greater than an order of magnitude of the cardinality of the data set from which the clustering model has been generated.

5 Conclusion

This paper described a novel methodology for network attack detection based on data mining of traffic information collected via SNMP by multiple monitoring stations, which are organized in a peer-to-peer network with the purpose of sharing the gained knowledge. In particular, the use of unsupervised clustering techniques on network-specific MIB objects allows to effectively detect malicious network behaviors, such as the ones due to DoS, DDoS and port scanning attacks, while still distinguishing between normal and harmful traffic profiles with very high accuracy.

Experimental results, obtained by emulating the real traffic of ten web servers under several kinds of attack, demonstrated the effectiveness of the proposed solution, reaching high accuracy levels with no detection failures and a false-positive rate as low as 1.21% on average. The accuracy levels of discerning normal and harmful traffic is on average greater than 99.58%. Moreover the detection accuracy can be increased by increasing the number of collaborative neighbours per peer, particularly the accuracy of identifying also the kind of attack.

Finally, the experiments highlighted that the loss of detection accuracy of not updated clustering models, over new incoming observations, is on average only 0.39%, after that the amount of the new SNMP traffic is an order of magnitude greater than the one used to learn the corresponding model.

Such promising results will be the basis to extend the current work to more complex network scenarios, where experiments will be conducted on SNMP traffic collected from a larger set of heterogeneous hosts and servers as well as from interconnecting equipment such as routers and switches.

References

1. Androulidakis, G., Chatzigiannakis, V., Papavassiliou, S.: Network anomaly detection and classification via opportunistic sampling. *IEEE Network* 23(1), 6–12 (2009)
2. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, pp. 91–99. Morgan kaufmann, San Francisco (1998)

3. Bridges, S.M., Vaughn, R.B.: Fuzzy data mining and genetic algorithms applied to intrusion detection. In: Proceedings of the National Information Systems Security Conference (NISSC), pp. 16–19 (2000)
4. Cabrera, J.B.D., Lewis, J.L., Qin, X., Lee, W., Mehra, R.K.: Proactive intrusion detection and distributed denial of service attacks—a case study in security management. *Journal of Network System Management* 10(2), 225–254 (2002)
5. CAIDA. The cooperative association for internet data analysis passive monitor (May 2009),
<http://www.caida.org/data/monitors/passive-equinix-chicago.xml>
6. Datta, S., Giannella, C.R., Kargupta, H.: Approximate distributed k-means clustering over a peer-to-peer network. *IEEE Transactions on Knowledge and Data Engineering* 21(10), 1372–1388 (2009)
7. Denning, D.E.: An intrusion-detection model. *IEEE Transactions on Software Engineering* 13(2), 222–232 (1987)
8. Dickerson, J.E., Dickerson, J.A.: Fuzzy network profiling for intrusion detection. In: Proc. of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, pp. 301–306 (2000)
9. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD 1996 Proceedings, pp. 226–231. AAAI Press, Menlo Park (1996)
10. Frawley, W.J., Piatetsky-shapiro, G., Matheus, C.J.: Knowledge discovery in databases: an overview. AAAI Press, Menlo Park (1992)
11. Ghoting, O.P., Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S.: Towards NIC-based intrusion detection. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 723–728. ACM Press, New York (2003)
12. Harrington, D., Presuhn, R., Wijnen, B.: An architecture for describing simple network management protocol (SNMP) management frameworks. IETF RFC 3411 (2002)
13. Hinneburg, A., Hinneburg, E., Keim, D.A.: An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD 1998), pp. 58–65. AAAI Press, Menlo Park (1998)
14. Johnson, E.L., Kargupta, H.: Collective, hierarchical clustering from distributed, heterogeneous data. In: Large-Scale Parallel KDD Systems, SIGKDD, pp. 221–244. Springer, Heidelberg (1999)
15. Kabiri, P., Ghorbani, A.A.: Research on intrusion detection and response: A survey. *International Journal of Network Security* 1, 84–102 (2005)
16. Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I.: On the capability of an SOM based intrusion detection system. In: Proceedings of the International Joint Conference on Neural Networks, July 2003, vol. 3, pp. 1808–1813 (2003)
17. Klusch, M., Lodi, S., Moro, G.: Distributed clustering based on sampling local density estimates. In: Proceedings of the Biennial International Joint Conference on Artificial Intelligence, pp. 485–490. Morgan Kaufmann, San Francisco (2003)
18. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
19. Mai, J., Sridharan, A., Chuah, C.-N., Zang, H., Ye, T.: Impact of packet sampling on portscan detection. *IEEE Journal on Selected Areas in Communications* 24(12), 2285–2298 (2006)

20. Monti, G., Moro, G.: Multidimensional range query and load balancing in wireless ad hoc and sensor networks. In: Wehrle, K., Kellerer, W., Singhal, S.K., Steinmetz, R. (eds.) *Peer-to-Peer Computing*, pp. 205–214. IEEE Computer Society, Los Alamitos (2008)
21. Moro, G., Ouksel, A.M.: G-grid: A class of scalable and self-organizing data structures for multi-dimensional querying and content routing in P2P networks. In: Moro, G., Sartori, C., Singh, M.P. (eds.) *AP2PC 2003*. LNCS (LNAI), vol. 2872, pp. 123–137. Springer, Heidelberg (2004)
22. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA 2001)*, pp. 5–8 (2001)
23. Costa Da Silva, J., Klusch, M., Lodi, S., Moro, G.: Privacy-preserving agent-based distributed data clustering. *Web Intelligence and Agent Systems* 4(2), 221–238 (2006)
24. Silverman, B.W.: *Density estimation for statistics and data analysis*. Chapman and Hall, London (1986)
25. Thottan, M., Ji, C.: Anomaly detection in IP networks. *IEEE Transactions on Signal Processing* 51(8), 2191–2204 (2003)
26. Vigna, G., Valeur, F., Kemmerer, R.A.: Designing and implementing a family of intrusion detection systems. *SIGSOFT Software Engineering Notes* 28(5), 88–97 (2003)
27. Xu, R., Wunsch II, D.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16(3), 645–678 (2005)
28. Xu, X., Ester, M., Kriegel, H.-P., Sander, J.: A distribution-based clustering algorithm for mining in large spatial databases. In: *Proceedings of the Fourteenth International Conference on Data Engineering (ICDE 1998)*, Washington, DC, USA, pp. 324–331. IEEE Computer Society, Los Alamitos (1998)
29. Yu, J., Lee, H., Kim, M.-S., Park, D.: Traffic flooding attack detection with SNMP MIB using SVM. *Computer Communications* 31(17), 4212–4219 (2008)
30. Zanero, S., Savaresi, S.M.: Unsupervised learning techniques for an intrusion detection system. In: *Proceedings of the 2004 ACM symposium on Applied Computing* (2004)
31. Zhang, R., Qian, D., Bao, C., Wu, W., Guo, X.: Multi-agent based intrusion detection architecture. In: *Proceedings of the 2001 International Conference on Computer Networks and Mobile Computing (ICCNMC 2001)*, Washington, DC, USA, p. 494. IEEE Computer Society, Los Alamitos (2001)
32. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, pp. 103–114 (1996)

A Scalable Approach to QoS-Aware Self-adaption in Service-Oriented Architectures

Valeria Cardellini¹, Emiliano Casalicchio¹, Vincenzo Grassi¹,
Francesco Lo Presti¹, and Raffaella Mirandola²

¹ Università di Roma “Tor Vergata”, Viale del Politecnico 1, 00133 Roma, Italy
{cardellini,casalicchio}@ing.uniroma2.it,
{vgrassi,lopresti}@info.uniroma2.it

² Politecnico di Milano, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
mirandola@elet.polimi.it

Abstract. In this paper we consider a provider that offers a SOA application implemented as a composite service to several users with different QoS requirements. For such a system, we present a scalable framework to the QoS-aware self-adaptation based on a two layer reference architecture. The first layer addresses the adaptation at the provisioning level: operating at a slower time scale, its role is to identify the set of candidate services to implement the system functionality at the required user QoS. The second layer addresses the adaptation at the service selection level: operating on a faster time scale, its role is to determine at running time the actual services which are bound to each user request while meeting both provider and user QoS. We formulate the adaptation strategy of both layers as suitable optimization problems which can be efficiently solved using standard techniques. Numerical experiments show the effectiveness of the proposed approach.

Keywords: Service-oriented architecture, self-adaptation, quality of service.

1 Introduction

The today increasingly complex software systems operating in a dynamic operational environment ask for management policies able to deal intelligently and autonomously with problems and tasks. Besides, the way software systems are developed is more and more based on the Service Oriented Architecture (SOA) paradigm, which encourages the construction of new applications through the identification, selection, and composition of network-accessible services offered by loosely coupled independent providers. In a “service market”, these different providers may offer different implementations of the same functionality (we refer to the former as *concrete services* and the latter as *abstract service*). These competing services are differentiated by their quality of service (QoS) and cost attributes, thus allowing a prospective user to choose the services that best suit

his/her needs. The QoS contracted by users and providers must meet certain respective obligations and performance expectations which the parties agree upon in the *Service Level Agreement* (SLA) contracts.

The fulfillment of global QoS requirements, such as the application response time and availability, by a SOA system offering a composite application is a challenging task, because it requires the system to take complex decisions within short time periods, in an operational environment characterized by a dynamic and unpredictable nature. A promising way to manage effectively this task is to make the SOA system able to self-adapt at runtime in response to changes in its operational environment, by autonomously reconfiguring itself through a closed-loop approach with feedback [1]. In this way, the system can timely react to environment changes (concerning for example available resources, type and amount of user requests), in such a way to fulfill its requirements at runtime.

Several methodologies have been already proposed for QoS-aware SOA systems able to dynamically self-adapt in order to fulfill non-functional or functional requirements (e.g., [2,3,4,5,6,7]). Most of the proposed methodologies address this issue as a *service selection* problem: given the set of abstract services needed to compose a new added value service, the goal is to identify for each abstract service a corresponding concrete service, selecting it from a set of candidates (e.g., [2,3,4,6,7]). When the operating conditions change (e.g., a selected concrete service is no longer available, or its delivered QoS has changed, or the user QoS requirements have changed), a new selection can be calculated and the abstract services which compose the offered SOA application are dynamically bound to a new set of concrete services.

In this paper, we follow the service selection approach towards self-adaptive SOA systems, but, differently from previous work in the area, we propose a two-layer adaptation strategy carried out by the service broker that offers the SOA application. In our approach, adaptation decisions occur at different time scales in order to exploit the optimal provisioning of the component services and maintain QoS guarantees to various classes of users. Specifically, the first layer operates at a slow time scale and addresses the adaptation task at the *service provisioning* level. Its role is to identify, from a given set of functionally equivalent candidate concrete services, the actual pool of concrete services that will be used to implement the component functionalities such that the aggregated QoS values satisfy the users' end-to-end QoS requirements and, at the same time, the service broker's utility function is maximized. The first layer also determines how much the identified concrete services are being utilized (i.e., it reserves the resource capacities). The solution provided by the first-layer is used on a long term for planning and defining SLAs with the service providers. The second layer operates at a fast time scale and addresses the adaptation at the *service selection* level. Its role is to determine, from the pool identified by the first layer, the actual concrete services which are bound to each incoming user request while meeting both provider and user QoS requirements.

We formulate the adaptation strategies of both layers as suitable optimization problems which can be solved using standard techniques. Specifically, the

second-layer optimization problem is formulated as a Linear Programming (LP) problem and is suitable to be solved at runtime because of its efficiency. On the other hand, the first-layer optimization problem is a Mixed Integer Linear Programming (MILP) one and is known to be NP-hard. However, its solution is required on a larger time scale than the second-layer problem: we estimate that, in a real scenario, the times at which the solution of two problems occurs differ by at least two orders of magnitude. Therefore, our two-layer approach can be deployed directly in a broker-based architecture operating in a highly variable SOA environment, where the scalability and effectiveness in replying to the users are important factors. To the best of our knowledge, this paper represents in the SOA environment the first proposal of a two-layer adaptation strategy operating at different time scales in order to manage dynamically the service provisioning and selection issues.

There is a significant body of research about how to realize the self-adaptation of systems to let them cope with a dynamic operational environment [1]. Existing proposals about how to architect a self-adaptive system share the common view that self-adaptation is achieved by means of a monitor-analyze-act cycle [8]: the system collects relevant events concerning itself and its context, analyzes them to decide suitable adaptation actions, and then act to execute the adaptation decisions. The main classes of approaches proposed in the SOA research community to tackle the dynamic adaptation of a SOA system include QoS-based service selection and workflow restructuring.

In the first case, as already outlined above, new service components are selected to deal with changes in the operating scenario [2,6,7,3,4,9,10,11]. Some of the works dealing with this general problem propose heuristics (e.g., [9,10] or genetic algorithms in [3]) to determine the adaptation actions. Others propose exact algorithms to this end: [6] formulates a multi-dimension multi-choice 0-1 knapsack problem as well as a multi-constraint optimal path problem; [7] presents a global planning approach to select an optimal execution plan by means of integer programming; in [2,10,11] the adaptation actions are selected through mixed integer programming. A general drawback of most proposals for dynamic adaptation based on service selection is that they pay little attention to efficiency and scalability. The approaches that we presented in [4,12] and adopt also in this paper address these issues by performing the optimization on a per-flow rather than per-request basis. In these approaches, the solution of the optimization problem holds for all the requests in a flow, and is recalculated only when some significant event occurs (e.g., a change in the availability or the QoS values of the selected concrete services). Moreover, the optimization problem is solved taking into account simultaneously the flows of requests generated by multiple classes of users, with possibly different QoS constraints.

The second class includes research efforts that have instead considered *workflow restructuring*, exploiting the inherent redundancy of the SOA environment to meet the QoS (basically, dependability) requirements [13,10,14]. In [12] we proposed a methodology that integrates within a unified framework the two classes of approaches by binding each abstract service to a set of functionally

equivalent concrete services, coordinated according to some spatial redundancy pattern. The two-layer approach we present in this paper can be extended by applying the above methodology.

The rest of the paper is organized as follows. In Section 2 we present the system architecture. In Section 3 we describe the composite service model we refer to, the type of SLA contracts used for the service users and providers, and define the goals of the two optimization problems. In Section 4 we present the mathematical formulation of the optimization problems used in the two-layer adaptation approach. In Section 5 we present the results of some numerical experiments. Finally, we draw some conclusions and give hints for future work in Section 6.

2 System Architecture

The *service broker* acts as a third-party intermediary between service users and providers, performing a role of provider towards the users and being in turn a requestor to the providers of the concrete services. It advertises and offers the composite service with a range of service classes which imply different QoS levels and monetary prices. To carry out its task, the broker architecture is structured around the following components, as illustrated in Figure 1: the *Workflow Engine*, the *Composition Manager*, the *SLA-P Manager*, the *Selection Manager*, the *SLA Monitor*, the *Optimization Engine*, the *Provisioning Manager*, and the *Data Access Library*. Our envisioned architecture is inspired by existing implementation of frameworks for Web services QoS brokering, e.g., [15,16].

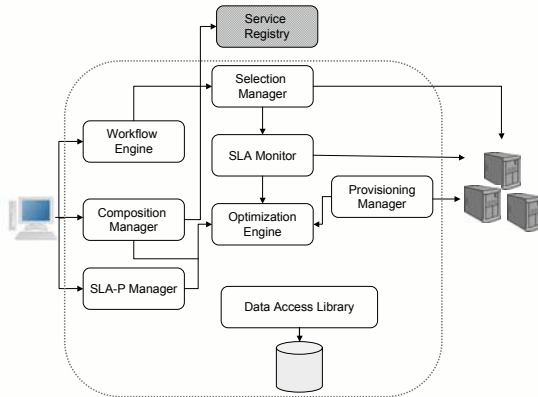


Fig. 1. Broker architecture

The respective tasks of the broker architecture components can be summarized as follows. The main functions of the Composition Manager are the specification of the business process and the discovery of all the service providers

offering functionally equivalent service implementations. The Workflow Engine is the software platform executing the BPEL business process (e.g., ActiveBPEL or ApacheODE) and represents the user front-end for the composite service provisioning. The Workflow Engine interacts with the Selection Manager to allow the invocation of the component services. Indeed, for each service invocation, the Selection Manager binds dynamically the request to the real endpoint that represents the concrete service. The latter is identified through the solution of the service selection optimization problem. Therefore, in the envisioned architecture the Selection Manager is in charge of the adaptation actions of the service selection layer. It also keeps up to date information about the composite service usage profile. Together, the Workflow Engine and the Selection Manager are responsible for managing the user requests flow, once the user has been admitted to the system with an established SLA.

The main task of the SLA-P Manager is the SLA negotiation with the users of the composite service. It is also in charge of the admission control and rate limiting functionalities. The first allows to determine whether a new user can be accepted, given the associated SLA and without violating already existing SLAs. To this end, the SLA-P Manager may trigger a new solution of the service selection problem. The rate limiting functionality is motivated by the need to limit the requests submitted to the composite service to the maximum arrival rate agreed in the SLAs. As a control mechanism for rate limiting, our broker architecture employs the classic token bucket [17]. This mechanism permits burstiness, but bounds it. The SLA-P Manager maintains a separate token bucket for each user, and each token in a bucket enables a single request to the composite service. Upon arrival, a request for the composite service will be sent out with the token bucket of the corresponding user decreased by one, provided there are available tokens for the request. Otherwise, the request is enqueued for subsequent transmission until tokens have been accumulated in the bucket. A SOA middleware architecture that employs the token-bucket algorithm for admission control is presented in [18].

The SLA Monitor collects information about the QoS level perceived by the users and offered by the providers of the used component services. Furthermore, the SLA Monitor signals whether there is some variation in the pool of service instances currently available for a given abstract service (i.e., it notifies if some service goes down/is unavailable).

The Optimization Engine is the broker component that executes the two adaptation algorithms (i.e, service provisioning and service selection), passing to them the updated instance of the optimization problem with the new values of the parameters. The calculated solutions provide indications about the adaptation actions that must be performed to identify the pool of resources (i.e., the concrete services) and to optimize their use with respect to the utility criterion of the broker as well as to the QoS levels agreed with the users.

The Provisioning Manager is in charge of organizing the service provisioning policy that makes the broker able to meet its utility objective, that is it manages the first-layer adaptation actions in the proposed system architecture. Once the

Optimization Engine has identified through the solution of the service provisioning problem the new subset of component services to be used, the Provisioning Manager negotiates the SLAs with their respective providers.

Finally, the Data Access Library is used by all the modules to access the model parameters of the composite service operations and environment (among which the abstract and the corresponding concrete services with their QoS values, and the values determined by the solution of the optimization problems, as discussed in Section 3). In Figure 1 the lines connecting the components to the Data Access Library have been omitted for clarity.

The SLA-P Manager, SLA Monitor, Selection Manager, and Composition Manager modules are collectively responsible for monitoring, detecting and deciding about the activation of a new adaptation strategy. When one of these modules detects a significant variation of the system model parameters, it signals the event to the Optimization Engine, which executes a new instance of the service provisioning or selection optimization problem and determines a new solution (in case it exists). Specifically, in our two-layer adaptation strategy the triggering to the Optimization Engine can occur either periodically or aperiodically and at different time scales. Given the efficiency of the service selection problem (formulated as LP problem in Section 4), it is suitable for being executed frequently in such a way to react quickly to detected changes. Its solution may be caused by a change in the effective request arrival rates measured by the SLA-P Manager at the exit of the token buckets, by a variation in the QoS levels determined by the SLA Monitor, and by a change in the composite service usage measured by the Selection Manager. If existing, the calculated solution provides indications to the Selection Manager on how to use the pool of available concrete services.

Since the first-layer service provisioning is a time-consuming reaction to detected changes (formulated as MILP problem in Section 4), it has to be invoked moderately and on a larger time scale. Its activation may be either periodic or aperiodic and it corresponds to modifications in the broker utility, in the arrival/departure of users, and also some change in the available resources (i.e., new concrete services identified by the Composition Manager, unreachability of some used concrete service determined by the SLA Monitor). The first-layer solution can be also triggered as a consequence of a second-layer optimization problem without a feasible solution. We postpone to a future paper the study of the possible activation schemes of the two layers and their performance impact on the system.

3 System Model

3.1 Composite Service Model

The SOA system managed by the broker offers a composite service, that is, a composition of multiple services in one logical unit in order to accomplish a complex task. We assume that the composite service structure is defined using BPEL [19], the de-facto standard for service workflows specification languages.

Here, without lack of generality, we restrict on the BPEL structured style of modeling, and consider workflows which include, besides the primitive `invoke` activity, all the different types of structured activities: `sequence`, `switch`, `while`, `pick`, and `flow`. Figure 2 shows an example of a BPEL workflow described as a UML2 activity diagram. With the exception of the `pick` construct, this example encompasses all the structured activities listed above.

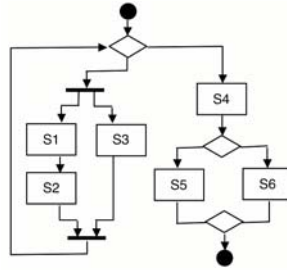


Fig. 2. An example of BPEL workflow

The business process for the composite service defines a set of abstract services \mathcal{V} . We denote by $S_i \in \mathcal{V}$ each abstract service (i.e., a functionality needed to compose a new added value service), and by $s_{ij} \in P_i$ a specific concrete service, where P_i is the set of functionally equivalent concrete services that have been identified by the Composition Manager as candidates to implement S_i . For each abstract service S_i , we also denote by $I_i \subseteq P_i$ the pool of concrete services determined by the solution of the service provisioning problem and used at runtime for offering the composite service.

The overall QoS of a composite service implementation depends not only on the QoS of the concrete services that have been bound to the abstract services and on the way they are orchestrated, but also on the usage profile of those services for each given class of users: a rarely invoked service has obviously a smaller impact on the overall QoS than a frequently invoked one, and different classes of users may invoke the same services with different frequencies. To embody this knowledge in our model, we model the usage profile of each service class $k \in K$ (where K denotes the set of the considered classes), by annotating each abstract service S_i with the average number of times V_i^k it is invoked by k -class requests addressed to the composite service. The Selection Manager performs a monitoring activity to keep up to date the V_i^k values.

3.2 SLA Model

Since the broker offering the composite service plays both the provider and requester roles, it is involved in two types of SLA, corresponding to these two roles: we call them SLA-P (provider role) and SLA-R (requester role). In general,

a SLA may include a large set of parameters, referring to different kind of QoS attributes (e.g., response time, availability, and reputation). In this paper, we restrict our attention to the following three attributes (but other attributes could be easily added to our framework without changing the methodology):

- *response time*: the interval of time elapsed from the service invocation to its completion;
- *availability*: the probability that the service completes its task when invoked;
- *cost*: the price charged for the service invocation.

The SLA-R contracted by the broker with the provider of the concrete service $s_{ij} \in I_i$ is specified by an instance of the tuple $\langle r_{ij}, a_{ij}, L_{ij}, c_{ij}, d_{ij} \rangle$, where r_{ij} and a_{ij} are the average response time and logarithm of availability of s_{ij} . In our SLA model, we assume that the price paid by the broker to the provider of s_{ij} is given by the sum of a fixed cost c_{ij} plus a variable cost, which is linearly proportional through d_{ij} to the amount of service capacity L_{ij} reserved by the broker. By solving the service provisioning optimization problem, the broker identifies the pool of concrete services with each of whom it negotiates an active SLA-R. The set of all the active SLAs-R defines the constraints within which the broker can organize the second stage of the adaptation strategy carried out through the service selection.

We denote by K the set of QoS classes offered by the broker. Each class $k \in K$ is characterized in terms of bounds on the expected response time R_{\max}^k and availability A_{\min}^k as well as the service costs: a fixed component c^k and a variable component which is proportional to a rate d^k per unit per request per unit of time. A user u requesting a given class of service k has to define the maximum load L_u^k it will generate. The SLA-P established by the broker with the requestor u for the QoS class k is therefore a tuple $\langle R_{\max}^k, A_{\min}^k, L_u^k, c^k, d^k \rangle$.

As discussed in Section 2, our broker architecture implements the token bucket mechanism for request rate limiting. The bucket of each user is refilled at rate L_u^k , until the bucket reaches its capacity. We denote by λ_u^k the effective arrival rate processed by the system.

3.3 Service Selection Model

The goal of the Selection Manager is to determine, for each QoS class, the concrete service s_{ij} that must be used to fulfill a request for the abstract service $S_i \in \mathcal{V}$. The selection can be modelled by associating with each S_i a vector $\mathbf{x}_i = (\mathbf{x}_i^1, \dots, \mathbf{x}_i^{|K|})$, where $\mathbf{x}_i^k = [x_{ij}^k]$ and $s_{ij} \in I_i$. Each entry x_{ij}^k of \mathbf{x}_i^k denotes the probability that the class- k request will be bound to the concrete service s_{ij} . With this model, we assume that the Selection Manager can probabilistically bind to different concrete services the requests (belonging to a same QoS class k) for an abstract service S_i . The deterministic selection of a single concrete service corresponds to the case $x_{ij}^k = 1$ for a given $s_{ij} \in I_i$.

As an example, consider the case $I_i = \{s_{i1}, s_{i2}, s_{i3}\}$ and assume that the adaptation policy x_i^k for a given class k specifies the following values: $x_{i1}^k = x_{i2}^k = 0.3$, $x_{i3}^k = 0.4$. This strategy implies that 30% of the class- k requests for service S_i are bound to service s_{i1} , 30% are bound to service s_{i2} while the remaining 40% are bound to s_{i3} . From this example we can see that, to get some overall QoS objective for a given class flow of requests, the Selection Manager may switch different requests to different providers (using x_i^k to drive the switch).

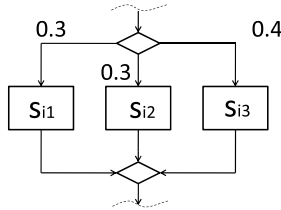


Fig. 3. Flow partitioning among different providers

The Selection Manager determines the values of the x_{ij}^k by invoking the Optimization Engine. The goal is to determine an overall selection strategy $\mathbf{x} = (x_1, \dots, x_{|\mathcal{V}|})$ which maximizes a suitable QoS objective function $F(\mathbf{x})$. The optimization problem takes the following general form:

find \mathbf{x} which maximizes $F(\mathbf{x})$
subject to: Class- k QoS due to strategy \mathbf{x}^k does not violate class- k SLA, $k \in K$;
 the load induced by strategy \mathbf{x} on provider s_{ij} does not exceed L_{ij} , $s_{ij} \in I_i$, $S_i \in \mathcal{V}$.

In our setting, the optimization problem takes the form of a LP problem. The details will be spelled out in Section 4.

3.4 Service Provisioning Model

The goal of the Provisioning Manager is to determine from the set of candidate concrete services the subset that will be used to implement the system functionalities and the capacity to be reserved on each selected concrete service. We model this selection with two vectors. The first vector is $\mathbf{y} = [y_{ij}]_{s_{ij} \in P_i, i \in \mathcal{V}}$, $y_{ij} \in \{0, 1\}$: $y_{ij} = 1$ if service $s_{ij} \in P_i$ is included in the pool I_i ; otherwise, $y_{ij} = 0$. We also define the vector $\mathbf{L} = [L_{ij}]_{s_{ij} \in P_i, i \in \mathcal{V}}$. L_{ij} is the service capacity the application reserves with the concrete service s_{ij} . $L_{ij} = 0$ if $y_{ij} = 0$ and $L_{ij} \geq 0$ if $y_{ij} = 1$.

The Provisioning Manager determines the values of the y_{ij} and L_{ij} by invoking the Optimization Engine. The goal is to determine the service pool and capacity which minimize a suitable cost function $C(\mathbf{y}, \mathbf{L})$. The optimization problem takes the following general form which we will detail in the next section:

find $(\mathbf{x}, \mathbf{y}, \mathbf{L})$ which minimizes $C(\mathbf{y}, \mathbf{L})$
subject to: Class- k QoS due to strategy \mathbf{x}^k does not violate class- k SLA, $k \in K$;
the service pool \mathbf{y} and capacity \mathbf{L} are such that the load
induced by strategy \mathbf{x} on provider s_{ij} does not exceed
 $L_{ij}, s_{ij} \in I_i, S_i \in \mathcal{V}$ for any possible class request arrival rate.

To understand the role of \mathbf{x} in this problem observe that for (\mathbf{y}, \mathbf{L}) to be feasible there must be at least one redirection strategy \mathbf{x} such that: 1) the load induced by \mathbf{x} on any provider does not exceed the capacity reserved on that provider for any class request arrival rate; and 2) the QoS of each class is not violated. On the other hand, we are not interested in optimizing (or even identifying) such strategy as long as one actually exists.

4 Optimization Problems

In this section, we first present how to compute the QoS attributes of the composite service. We then detail the instances of the optimization models we presented in Section 3.

4.1 QoS Metrics

For each class $k \in K$ offered by the broker, the overall QoS attributes are the expected response time R^k and the expected availability A^k . To compute these quantities, let $Z_i^k(\mathbf{x})$, $Z \in \{R, A\}$, denote the QoS attribute of the abstract service $S_i \in \mathcal{V}$. We have $Z_i^k(\mathbf{x}) = \sum_{s_{ij} \in I_i} x_{ij}^k z_{ij}^k$ where $z_{ij}^k, z \in \{r, a\}$ is the corresponding QoS attribute offered by the concrete service s_{ij} which can implement S_i . We now derive closed form expressions for the QoS attributes of the composite service we will later use in the formulation of the optimization problem.

Availability. The (logarithm of the) availability QoS metric is an additive metric [20]. Therefore, for its expected value we readily obtain

$$A^k(\mathbf{x}) = \sum_{i \in \mathcal{V}} V_i^k A_i^k(\mathbf{x}) = \sum_{i \in \mathcal{V}} V_i^k \sum_{s_{ij} \in I_i} x_{ij}^k a_{ij}$$

where V_i^k is the expected number of times S_i is invoked for a class- k request.

Response Time. The response time metric is additive only as long as the composite service does not include flow structured activities. In such cases, we readily have:

$$R^k(\mathbf{x}) = \sum_{i \in \mathcal{V}} V_i^k \sum_{s_{ij} \in I_i} x_{ij}^k r_{ij}. \tag{1}$$

In the general case, instead, we need to account for the fact that the response time of a flow activity [19] is given by the largest response time among its component activities. Hence, in the general case, the response time is not additive and (1) does not hold. In this case, we derive an expression for the response time $R^k(\mathbf{x})$ by recursively computing the response time of the constituent workflow activities as shown in [4] which we will later use in the actual problem formulation.

4.2 Second-Layer Problem: Service Selection Optimization

In this section we detail the service selection optimization problem. The goal is to determine the variables x_{ij}^k , $i \in \mathcal{V}$, $k \in K$, $s_{ij} \in I_i$ which maximize a suitable QoS function. We assume that the broker wants, in general, to optimize multiple QoS attributes (which can be either mutually independent or possibly conflicting), rather than just a single one, *i.e.*, the response time. We thus consider as objective function $F(\mathbf{x})$ an aggregate QoS measure given by a weighted sum of the (normalized) QoS attributes. More precisely, let $Z(\mathbf{x}) = \frac{1}{\sum_{k \in K} \lambda^k} \sum_{k \in K} \lambda^k Z^k(\mathbf{x})$, where $Z \in \{R, A\}$ is the expected overall response time and availability, respectively, and $\lambda^k = \sum_u \lambda_u^k$ is the instantaneous aggregate flow of class- k requests. We define the objective function as follows:

$$F(\mathbf{x}) = w_r \frac{R_{\max} - R(\mathbf{x})}{R_{\max} - R_{\min}} + w_a \frac{A(\mathbf{x}) - A_{\min}}{A_{\max} - A_{\min}} \quad (2)$$

where $w_r, w_a \geq 0$, $w_r + w_a = 1$, are weights for the different QoS attributes. R_{\max} (R_{\min}), and A_{\max} (A_{\min}) denote, respectively, the maximum (minimum) value for the overall response time and the (logarithm of) availability. We will describe how to determine these values shortly.

The Optimization Engine task consists in finding the variables x_{ij}^k , $i \in \mathcal{V}$, $k \in K$, $s_{ij} \in I_i$, which solve the following optimization problem:

$$\mathbf{max} \quad F(\mathbf{x})$$

$$\mathbf{subject\ to:} \quad R^k(\mathbf{x}) \leq R_{\max}^k \quad k \in K \quad (3)$$

$$R_{l'}^k(\mathbf{x}) \leq R_l^k(\mathbf{x}) \quad l' \in d(l), l \in \mathcal{F}, k \in K \quad (4)$$

$$R_l^k(\mathbf{x}) = \sum_{i \in \mathcal{V}, i \prec_{ddl} l} \frac{V_i^k}{V_l^k} \sum_{s_{ij} \in I_i} x_{ij}^k r_{ij} + \sum_{h \in \mathcal{F}, h \prec_{ddl} l} \frac{V_h^k}{V_l^k} R_h^k(\mathbf{x}), l \notin \mathcal{F}, k \in K \quad (5)$$

$$A^k(\mathbf{x}) \geq A_{\min}^k \quad k \in K \quad (6)$$

$$\sum_{k \in K} x_{ij}^k V_i^k \lambda^k \leq L_{ij} \quad i \in \mathcal{V}, s_{ij} \in I_i \quad (7)$$

$$x_{ij}^k \geq 0, \quad s_{ij} \in I_i, \quad \sum_{s_{ij} \in I_i} x_{ij}^k = 1 \quad i \in \mathcal{V}, k \in K \quad (8)$$

Equations (3)-(6) are the QoS constraints for each service class on response time and availability, where R_{\max}^k and A_{\min}^k are respectively the maximum response time and the minimum (logarithm of the) availability that characterize the QoS class k . The constraints for the response time take into account the fact the response time of a flow activity is given by the largest response time of its component activities. This is reflected in the constraints (4)-(5), where \mathcal{F} denotes the set of flow activities in the composite service. Inequalities (4), in particular,

allow us to express the relationship among the response time R_l^s of a flow activity and that of its component activities R_p^s . For each flow activity l , $d(l)$ is the set of top-level activities/services which are nested within l ; $i \prec_{dd} l$ means that service i occurs within activity j in the BPEL code and, within j , i does not appear within a flow activity (see [4] for details). Equations (7) are the SLA-R constraints and ensure that the application does not exceed the volume of invocations agreed with the service providers. Finally, Equations (8) are the functional constraints.

The maximum and minimum values of the QoS attributes in the objective function (2) are determined as follows. R_{\max} and A_{\min} are simply expressed respectively in terms of R_{\max}^k and A_{\min}^k . For example, the maximum response time is given by $R_{\max} = \frac{1}{\sum_{k \in K} \lambda^k} \sum_{k \in K} \lambda^k R_{\max}^k$. Similar expression holds for A_{\min} . The values for R_{\min} and A_{\max} , instead, are determined by solving a modified optimization problem in which the objective function is the QoS attribute of interest, subject to the constraints (7)-(8).

We observe that the proposed Optimization Engine problem is a Linear Programming problem which can be efficiently solved via standard techniques. The solution thus lends itself to on-line operations.

4.3 First-Layer Problem: Service Provisioning Optimization

We now turn our attention to the Provisioning Manager optimization problem. The goal is to determine the value of the variables y_{ij} and L_{ij} , $s_{ij} \in P_i$, $i \in \mathcal{V}$, which minimize the broker cost function. We consider as objective function $F(\mathbf{y}, \mathbf{L})$ the following simple cost function:

$$F(\mathbf{y}, \mathbf{L}) = \sum_{s_{ij} \in P_i, i \in \mathcal{V}} c_{ij} y_{ij} + d_{ij} L_{ij} \tag{9}$$

where c_{ij} represents a fixed/flat cost to be paid for using concrete service s_{ij} and d_{ij} is the cost for unit of capacity of service s_{ij} , reserved by the broker.

The Optimization Engine task consists in finding the y_{ij} and L_{ij} , $s_{ij} \in P_i$, $i \in \mathcal{V}$ (and also x_{ij}^k , $s_{ij} \in P_i$, $i \in \mathcal{V}$, $k \in K$) which solve the following optimization problem:

$$\begin{aligned} & \min F(\mathbf{y}, \mathbf{L}) \\ & \text{subject to:} \end{aligned} \tag{10}$$

$$\text{QoS constraints (3) - (6)}$$

$$\sum_{k \in K} x_{ij}^k V_i^k L^k \leq L_{ij} \quad i \in \mathcal{V}, s_{ij} \in P_i, \tag{11}$$

$$L_{ij} \leq M_{ij} y_{ij}, \quad i \in \mathcal{V}, s_{ij} \in P_i \tag{12}$$

$$x_{ij}^k \leq y_{ij} \quad i \in \mathcal{V}, k \in K \tag{13}$$

$$x_{ij}^k \geq 0, s_{ij} \in P_i, \sum_{s_{ij} \in P_i} x_{ij}^k = 1 \quad i \in \mathcal{V}, k \in K \tag{14}$$

$$y_{ij} \in \{0, 1\}, \quad i \in \mathcal{V}, s_{ij} \in P_i \tag{15}$$

The constraints (3)-(6) are the QoS constraints as in the service selection optimization. The constraints (11) are the provider capacity constraints which require that the reserved capacity $L_{ij}, s_{ij} \in P_i, i \in \mathcal{V}$ must accommodate any request load for the concrete service s_{ij} (under service selection strategy \mathbf{x}), where $L^k = \sum_u L_u^k$ denotes the maximum class k request rate. Finally, equations (12)-(14) are the functional constraints. (12) requires that $y_{ij} = 1$ for L_{ij} be greater than 0; similarly, (13) requires $y_{ij} = 1$ for x_{ij}^k be greater than 0. In (11) we also introduce the constant M_{ij} which denotes the maximum capacity that can be reserved on provider s_{ij} (M_{ij} thus captures the finiteness of provider s_{ij} resources).

The proposed optimization problem is a MILP problem. It is known to NP-hard with the complexity being exponential in the number of integer variables, which is $O(\max_{i \in \mathcal{V}} |P_i| \times |\mathcal{V}|)$.

5 Numerical Experiments

In this section, we illustrate the behaviour of the proposed two-layer adaptation strategy through the simple abstract workflow of Figure 2. We consider a broker which offers two QoS classes *gold* and *silver*, denoted by the superscript 1 and 2, respectively. Table 1 summarizes the two classes QoS attributes. The gold class guarantees to its users low response times at a high cost, while the silver class offers a cheaper alternative with higher response times. We consider the following values for the the number of service invocations: $V_1^k = V_2^k = V_k^3 = 1.5$, and $V_4^k = 1$ for $k = 1, 2, V_5^1 = 0.7, V_6^1 = 0.3$, and $V_5^2 = V_6^2 = 0.5$. We assume that for each abstract service there are four providers which implements it. The concrete services differ in terms of response time, cost and availability. Table 2 summarizes their system parameters. They have been ordered so that for each abstract service $S_i \in \mathcal{V}$, s_{ij} represents the *better*, albeit more expensive, service, with respect $s_{ij'}$, $j' > j$. For all services, we assume $M_{ij} = 10$.

We study now the broker behaviour over a period of 1000 time units during which we have a fixed set of users. The associated peak request rate for the two service classes is assumed equal to $(L^1, L^2) = (4, 7)$. We assume that during this period the only meaningful event is the unavailability of service s_{14} from time 400 onward. First we consider the behaviour of the Provisioning Manager. The role of the manager is to identify the optimal - cost wise - set of concrete services to implement the abstract services and the associated capacities. For the given workflow, the solution of the optimization problem is illustrated in Table 3, which reports the set I_i of concrete services selected for each abstract service

Table 1. Composite service class attributes

QoS Class	R_{\max}^k	A_{\min}^k	c^k	d^{jk}
<i>gold</i>	12	$\log(0.95)$	10	5
<i>silver</i>	20	$\log(0.9)$	6	3

Table 2. Concrete services QoS attributes

s_{ij}	r_{ij}	a_{ij}	c_{ij}	d_{ij}
s_{11}	1	$\log(0.999)$	3	2
s_{12}	1.5	$\log(0.995)$	4	1.5
s_{13}	1.5	$\log(0.99)$	3	1.5
s_{14}	3.5	$\log(0.98)$	2.5	1
s_{21}	2	$\log(0.999)$	4	1.5
s_{22}	4	$\log(0.99)$	2	1.5
s_{23}	1	$\log(0.99)$	4.5	1
s_{24}	5	$\log(0.95)$	1	1
s_{31}	1	$\log(0.999)$	4	1.5
s_{32}	1	$\log(0.99)$	2	1.5
s_{33}	2	$\log(0.99)$	4.5	1
s_{34}	3	$\log(0.99)$	1	1
s_{41}	0.5	$\log(0.999)$	0.6	2
s_{42}	1	$\log(0.995)$	0.5	1
s_{43}	1	$\log(0.99)$	0.4	1.5
s_{44}	2	$\log(0.99)$	0.3	1
s_{51}	2	$\log(0.999)$	1	2
s_{52}	2	$\log(0.995)$	0.7	1
s_{53}	2.2	$\log(0.99)$	0.5	1.5
s_{54}	3	$\log(0.99)$	0.2	1.5
s_{61}	1.8	$\log(0.999)$	0.5	1.5
s_{62}	2	$\log(0.995)$	0.4	1
s_{63}	2	$\log(0.99)$	0.3	1
s_{64}	4	$\log(0.99)$	0.2	1.5

$S_i \in \mathcal{V}$ and the capacity L_{ij} reserved in each concrete service. A first solution (Table 3 (left)) is first computed at the beginning of the period (for a minimum cost equal to 93.8). The solution guarantees enough resources to sustain peak rate traffic, *i.e.*, $(L^1, L^2) = (4, 7)$ at the required QoS of each class. Observe that since the different abstract services are characterized by different frequencies of invocations, the overall capacity to be reserved differs from service to service, *e.g.*, S_1 requires an overall capacity of 16.5, while S_5 requires only a capacity of 6.3. At time 400, we assume that service s_{14} becomes unavailable. In our example, this forces the Provisioning Manager to execute again the provisioning optimization problem (it is not possible to serve the requests for the abstract service S_1 with the sole concrete service s_{13}) and adjusts the SLA with the providers accordingly. Table 3 (right) shows the new solution where, essentially, the concrete service s_{13} replaces s_{14} and the reserved capacity of some providers are slightly modified.

Table 3. SLA Manager solution. Service pool and reserved capacities.

Service Sets	Reserved Capacity
$I_1 = \{s_{13}, s_{14}\}$	$L_{13} = 6.5, L_{14} = 10$
$I_2 = \{s_{23}, s_{24}\}$	$L_{23} = 9.2, L_{24} = 7.8$
$I_3 = \{s_{32}, s_{34}\}$	$L_{32} = 10, L_{34} = 6.5$
$I_4 = \{s_{42}, s_{44}\}$	$L_{42} = 6.9, L_{44} = 4.1$
$I_5 = \{s_{52}\}$	$L_{52} = 6.3$
$I_6 = \{s_{63}\}$	$L_{63} = 4.7$

Service Sets	Reserved Capacity
$I_1 = \{s_{12}, s_{13}\}$	$L_{12} = 8.35, L_{13} = 8.15$
$I_2 = \{s_{23}, s_{24}\}$	$L_{23} = 8.93, L_{24} = 7.57$
$I_3 = \{s_{32}, s_{34}\}$	$L_{32} = 10, L_{34} = 6.5$
$I_4 = \{s_{42}, s_{44}\}$	$L_{42} = 6.36, L_{44} = 4.64$
$I_5 = \{s_{52}\}$	$L_{52} = 6.3$
$I_6 = \{s_{63}\}$	$L_{63} = 4.7$

We now turn our attention to the Selection Manager. Differently from the Provisioning Manager, the Selection Manager adaptation role is to determine at running time the actual services to be bound to each user request. To illustrate its behaviour we consider the sample path arrival rates for the two classes shown in Figure 4 (the sample paths have been generated by superposition of several regulated sources, with each source being a two state on-off source). We assume

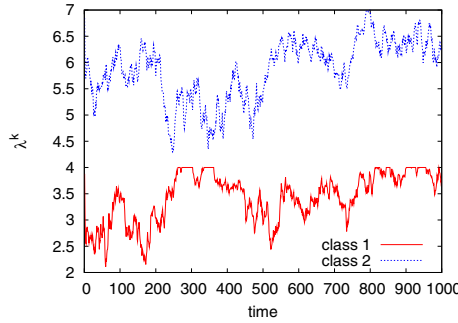


Fig. 4. Sample path arrival rate λ^k

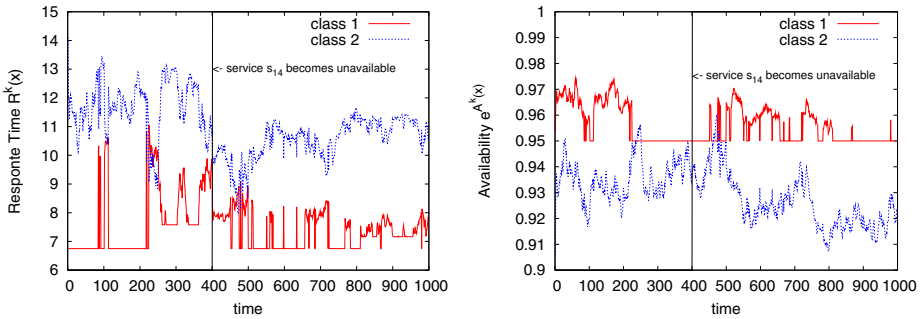


Fig. 5. Selection Manager solution. QoS metrics: response time $R^k(\mathbf{x})$ (left); Availability $e^{A^k(\mathbf{x})}$ (right).

that the Selection Manager uses the measured actual aggregate arrival rates $\lambda^k \leq L^k$, $k = 1, 2$ and solves the service selection optimization problem to settle the vector \mathbf{x} , according to which randomly determines the concrete service to select. Different values of λ^k , $k = 1, 2$ result into different optimal vectors which in turn yield different QoS metrics.

In Figure 5 we show how the expected composite service QoS metrics vary over time for the two classes under the assumption the Selection Manager minimizes the service response time, *i.e.*, $w_r = 1$. Both service response time and availability vary with the request rates but are always within the performance bound defined by the class SLA metrics. Not surprisingly, users experience better response time and service availability for lower request rates since a large fraction - if not all - of the requests are bound to the best services in the pool. Observe that after $t=400$, the response time for both service classes improves significantly. This can be explained by observing that the unavailability of service s_{14} , which provides the cheapest - but slowest - service, forces the broker to include in the pool the more expensive, but faster, service s_{12} , which results into overall better response times.

6 Conclusions

This paper deals with a two-layer approach for QoS-aware adaptation of SOA systems. The basic guideline we have followed in its definition has been to devise an adaptation strategy that is efficient and scalable to make realistic its use in taking runtime decisions in a rapidly changing environment. This efficiency is achieved by decomposing the service provisioning and service selection optimizations into two independent phases occurring at different time scales. The service selection problem can be solved on a fast time scale at each detected significant change which stems from the system's self or context. The sustainable frequent rate of solution derives from the formulation as a constrained optimization problem that can be efficiently solved via standard techniques and tools for linear programming. The more time consuming service provisioning problem can be solved on a slower time scale because it addresses the identification of the pool of concrete services to be used by the broker for the SLA management with the service providers. Besides being efficient, the proposed approach is also flexible, because it can be simultaneously used to serve the requests of multiple classes of users.

Our future work will address the issues concerning the implementation of the two-layer adaptation approach, such as the temporal aspects of change (e.g., the monitoring and detection of significant changes that trigger the decision on what needs to be changed). The implementation of a system prototype we are currently working on will allow us to validate the proposed approach through a real set of experiments.

Acknowledgments

This work is supported by the Italian PRIN 2007 project "D-ASAP: Dependable Adaptable Software Architectures for Pervasive Computing".

References

1. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.* 4(2), 1–42 (2009)
2. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. *IEEE Trans. Softw. Eng.* 33(6), 369–384 (2007)
3. Canfora, G., Penta, M.D., Esposito, R., Villani, M.L.: A framework for qos-aware binding and re-binding of composite web services. *J. Syst. Softw.* 81(10), 1754–1769 (2008)
4. Cardellini, V., Casalicchio, E., Grassi, V., Lo Presti, F.: Flow-based service selection for web service composition supporting multiple qos classes. In: *ICWS 2007*, pp. 743–750. IEEE Computer Society, Los Alamitos (2007)
5. Maximilien, E.M., Singh, M.P.: Toward autonomic web services trust and selection. In: *ICSOC 2004*, pp. 212–221. ACM, New York (2004)
6. Yu, T., Zhang, Y., Lin, K.J.: Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Trans. Web* 1(1), 1–26 (2007)

7. Zeng, L., Benatallah, B., Dumas, M., Kalagnamam, J., Chang, H.: QoS-aware middleware for web services composition. *IEEE Trans. Soft. Eng.* 30(5) (2004)
8. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. *IEEE Computer* 36(1), 41–50 (2003)
9. Menascé, D.A., Casalicchio, E., Dubey, V.: On optimal service selection in service oriented architectures. *Perform. Eval.* (2009)
10. Guo, H., Huai, J., Li, H., Deng, T., Li, Y., Du, Z.: Angel: Optimal configuration for high available service composition. In: *ICWS 2007*, pp. 280–287. IEEE Computer Society, Los Alamitos (2007)
11. Qu, Y., Lin, C., Wang, Y., Shan, Z.: Qos-aware composite service selection in grids. In: *GCC 2006*, pp. 458–465. IEEE Computer Society, Los Alamitos (2006)
12. Cardellini, V., Casalicchio, E., Grassi, V., Lo Presti, F., Mirandola, R.: Qos-driven runtime adaptation of service oriented architectures. In: *ESEC/FSE 2009*, pp. 131–140. ACM, New York (2009)
13. Chafle, G., Doshi, P., Harney, J., Mittal, S., Srivastava, B.: Improved adaptation of web service compositions using value of changed information. In: *ICWS 2007*, pp. 784–791. IEEE Computer Society, Los Alamitos (2007)
14. Stein, S., Payne, T.R., Jennings, N.R.: Flexible provisioning of web service workflows. *ACM Trans. Internet Technol.* 9(1), 1–45 (2009)
15. Menascé, D., Ruan, H., Gooma, H.: QoS management in service oriented architectures. *Perform. Eval.* 7-8(64) (2007)
16. Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web services on demand: WSLA-driven automated management. *IBM Systems J.* 43(1) (2004)
17. Tang, P., Tai, C.: Network traffic characterization using token bucket model. In: *IEEE Infocom 1999* (1999)
18. Liu, Y., Tan, M., Gorton, I., Clayphan, A.J.: An autonomic middleware solution for coordinating multiple qos controls. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) *ICSOC 2008*. LNCS, vol. 5364, pp. 225–240. Springer, Heidelberg (2008)
19. OASIS: Web Services Business Process Execution Language Version 2.0 (2007), <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
20. Cardoso, J., Sheth, A.P., Miller, J.A., Arnold, J., Kochut, K.J.: Modeling quality of service for workflows and web service processes. *Web Semantics J.* 1(3) (2004)

QShine 2009

Invited Session III – QoS and Power Consumption

Throughput and Energy Efficiency in IEEE 802.11 WLANs: Friends or Foes?

Pablo Serrano¹, Albert Banchs¹, Luca Vollero², and Matthias Hollick³

¹ Universidad Carlos III de Madrid, 28911, Leganés, Spain
pablo@it.uc3m.es, banchs@it.uc3m.es

² Università Campus Bio-Medico di Roma, 00128, Roma, Italy
l.vollero@unicampus.it

³ Technische Universität Darmstadt, 64283, Darmstadt, Germany
matthias.hollick@cased.de

Abstract. Understanding and optimizing the energy consumption of wireless devices is critical to maximize network lifetime and to provide guidelines for the design of new protocols and interfaces. In this work we first provide an accurate analysis of the energy performance of an IEEE 802.11 WLAN, and then we derive the configuration to maximize it. We also analyze the impact of the energy configuration of the device on the throughput performance, and discuss in which circumstances throughput and energy efficiency can be both maximized and where they constitute different challenges.

Keywords: Energy efficiency, energy optimization, throughput optimization, IEEE 802.11.

1 Introduction

ICT technologies hold one of the keys to the reduction of greenhouse gases produced worldwide. The importance of “greening of the Internet” is thus recognized as a primary design goal of future global network infrastructures. It is estimated that, today, the Internet already accounts for about 2% of total world energy consumption, and with the current trend of shifting offline services online, this percentage is expected to grow significantly in the next years. The energy consumption is to be further fuelled by the forthcoming Internet-based platforms that require always-on connectivity.

However, communication protocols, and in particular the technologies used in the access network, have been originally conceived to optimize metrics other than energy, such as throughput or delay. *Greening* these protocols thus represents a shift in the design paradigm, where energy instead of time is the most critical network resource. We no longer want to maximize the bits sent per time unit, but instead the bits the network can send per each joule consumed. Still, it is clear that this comes not for free, and there is a price to pay when developing sustainable architectures.

In this paper we assess to which extent the (old) throughput-maximization and the (new) efficiency-maximization objectives diverge, for the case of 802.11

WLANs. Previous work has solved the configuration of WLANs for throughput maximization, starting from the static approaches of [2,10] and including later adaptive approaches to maximize the bits per second sent [7]. However, from the point of view of energy consumption, most of the research so far has addressed the analytical or experimental characterization of the energy consumption of the WLAN [9,5,6], which is typically divided in three states: transmission, reception and idle-state (see Table 1 for the energy consumption of selected wireless network cards). There has been also some proposals for efficiency optimization (e.g. [14,8]), typically based on heuristic and sometimes requiring changes to the MAC layer. To the best of our knowledge, only Bruno et al. [3] have considered the relation between throughput and energy and have discussed whether they could be both jointly maximized or not. In their model, consisting of a p-persistent CSMA-based WLAN where interfaces only consumed energy in two states (transmission and reception), the answer was yes. In this paper, where we improve the accuracy of the consumption model, we prove that this is not always the case.

The rest of the paper is organized as follows. In Section 2 we present and validate an analytical model of the energy consumption of a WLAN. We further introduce a new *approximate* model that trades off accuracy for the sake of simplicity (nevertheless, as shown in the validation part, this reduction of accuracy is negligible). Section 3 presents the two approaches for performance maximization: the throughput-based approach of Bianchi, and our energy-based approach that builds upon the approximate analysis to derive a closed-form expression for the optimal transmission probability. In Section 4 we compare the resulting configuration and performance from each approach, while Section 5 concludes the paper.

2 Energy Consumption Analysis

Our analytical model for the consumption of a WLAN requires the following input parameters: N , the number of stations in the WLAN. W , defined as the minimum contention window stations use on their first attempt, and $\{\rho_t, \rho_r, \rho_i\}$, defined as the power consumed by the wireless interfaces when transmitting, receiving or idling. We assume all stations have always a packet of fixed length L ready for transmission, i.e., the network operates under saturation conditions, and that the sole reason for frame loss is a collision (where two or more stations transmit simultaneously). We further assume that each station randomly selects the destination for each frame out of the other $N - 1$ stations.

2.1 Model

With the assumption that each transmission attempt collides with a constant and independent probability, we can model the behavior of a station with the same Markov chain used in [2]. Then, the probability that a station operating

under saturation conditions transmits upon a backoff counter decrement can be computed by means of the following equation given by [2]

$$\tau = \frac{2}{1 + W + pW \sum_{i=0}^{m-1} (2p)^i}$$

where p is the probability that a transmission attempt of a station collides. This probability can be computed as

$$p = 1 - (1 - \tau)^{N-1}$$

The above constitutes a system of two non-linear equations that can be solved numerically, giving the value for τ . With this, we next proceed to compute the energy per slot consumed by a station, which we denote by e .

We compute e by applying the total probability theorem as follows:

$$e = \sum_{j \in \Theta} E(j)p(j) \tag{1}$$

where Θ is the set of events that can take place in a single timeslot, while $E(j)$ and $p(j)$ are the energy consumed in case of event j given its probability, respectively. The set Θ contains the following events, along with their probabilities:

- The slot is empty, p_e
- There is a success from the considered station, $p_{s,i}$
- There is a success from another station, $p_{s,-i}$
- There is a collision and the considered station is involved, $p_{c,i}$
- There is a collision but the considered station is not involved, $p_{c,-i}$

This way we can expand (1) with these probabilities and the energy consumed per event can be derived as follows:

$$\begin{aligned} e = & p_e \rho_i T_e + \\ & + p_{s,i} (\rho_t T_s + \rho_r T_{ack} + \rho_i (SIFS + DIFS)) + \\ & + p_{s,-i} \left[\rho_r T_s + \frac{1}{N-1} (\rho_t T_{ack}) + \right. \\ & \left. + \frac{N-2}{N-1} \rho_r (\rho_r T_{ack}) + \rho_i (SIFS + DIFS) \right] + \\ & + p_{c,i} (\rho_t T_s + \rho_i EIFS) + p_{c,-i} (\rho_r T_s + \rho_i EIFS) \end{aligned}$$

where T_e , T_s , and T_{ack} are the durations of an empty slot, a successful transmission and the transmission of an acknowledgment, while $SIFS$, $DIFS$, and $EIFS$ are physical constants (for the computation of these values, see e.g. [2]).

The probability of each event can be easily computed based on the probability of a transmission τ as follows

$$\begin{aligned}
p_e &= (1 - \tau)^N \\
p_s &= N\tau(1 - \tau)^{N-1} \\
p_{s,i} &= \tau(1 - \tau)^{N-1} \\
p_{s,\neg i} &= p_s - p_{s,i} \\
p_c &= 1 - p_e - p_s \\
p_{c,i} &= \tau(1 - (1 - \tau)^{N-1}) \\
p_{c,\neg i} &= p_c - p_{c,i}
\end{aligned}$$

However, note that the full expression of (11) consists of a sum of several terms that non-linearly depends on τ . In order to derive the value of τ that provides the best energy performance, we introduce the following simplified expression for e

$$\hat{e} = (1 - \tau)^N \rho_e T_e + \tau \rho_t T_s + (1 - \tau) (1 - (1 - \tau)^N) \rho_r T_s$$

This way, we have simplified the set Θ of events by considering only three cases: *i*) nobody transmits, *ii*) the station transmits (without the distinction if there is a collision or a success), and *iii*) someone else transmits (again, no matter if there is a success of a collision).

The above can be expressed as:

$$\hat{e} = R + \tau(T - R) - (1 - \tau)^N(R - E)$$

where $E = \rho_e T_e$, $T = \rho_t T_s$, and $R = \rho_r T_s$. We further write $T' = T - R$ and $R' = R - E$, therefore:

$$\hat{e} = R + \tau T' - (1 - \tau)^N R' \quad (2)$$

(Note that in the following section we assess the accuracy obtained both via (11) and (2).) Finally, we define the energy efficiency η as the ratio between the bits transmitted and the energy consumed in a timeslot:

$$\eta = \frac{p_{s,i} L}{e} \quad (3)$$

2.2 Validation

We first compare the accuracy of the exact and approximate models for e and \hat{e} versus results obtained via simulation. To this end, we compare the energy consumed per timeslot for the three selected power consumption sets listed in Table 1 for different values of N and the default DCF configuration. Results are shown in Fig. 1.

From the results, it is clear that the detailed analytical model e provides values that almost coincide with those derived from simulations, while the approximate model \hat{e} follows quite closely the behavior of the WLAN but slightly overestimating the energy consumed for large values of N .

Table 1. Power consumption in Watts for different wireless interfaces (as reported in [1])

Card	ρ_t	ρ_r	ρ_i
Lucent WaveLan (A)	1.650	1.400	1.150
SoketCom Compact Flash (B)	0.924	0.594	0.066
Intel PRO 2200 (C)	1.450	0.850	0.080

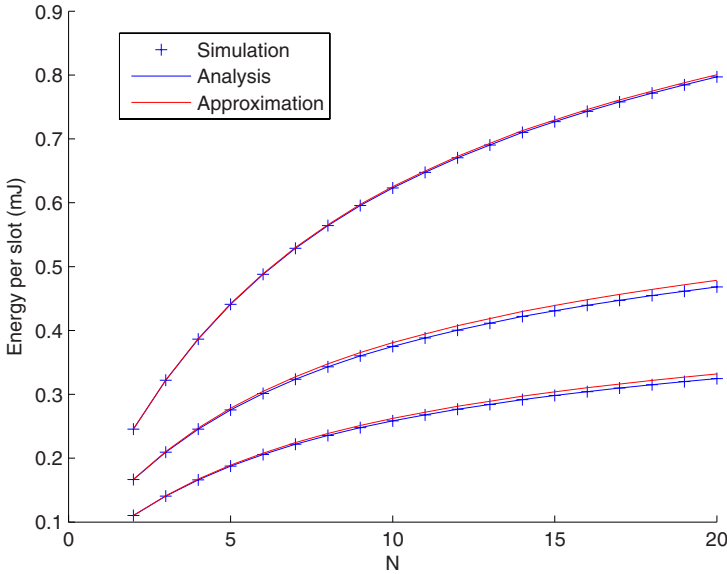


Fig. 1. Energy per slot-time consumed for different interfaces and number of stations. The arrays of curves from top to bottom show the results for energy profiles (A), (C), and (B).

We take advantage of the accurate analytical model to further explore the energy consumption of the WLAN, identifying where is the energy consumed. To this aim, we account for the relative amount of energy spent on successful transmissions, collisions and idling, with the results of Fig. 2 for the case of $N = 10$ and the interface A of Table 1.

As can be seen from the figure, it is clear that for relatively small values of CW_{min} there is a lot of energy wasted in collisions, while the energy spent idling is quite small. Then, with increasing CW_{min} values the energy wasted in collisions decreases rapidly, while there is a slower increase in the part corresponding to idling. This behavior is intuitively explained as follows. Increasing CW_{min} results in a smaller collision probability and larger probability of empty timeslots. However, the savings in energy due to the absence of collisions are “multiplied” by the power consumption when receiving ρ_r or transmitting ρ_t as

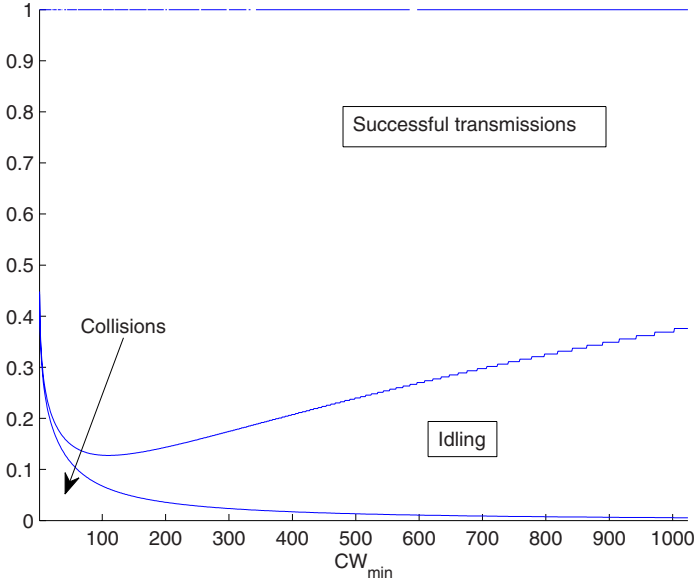


Fig. 2. Relative energy devoted to successful transmissions, collisions and idling

well as, approximately, the length of a successful transmission T_s . On the other hand, the increase of energy consumption because of the larger number of empty timeslots is weighted by ρ_i and T_e , both being smaller than their counterparts.

Another result from the figure is that there exists a maximum for the energy devoted to successful transmission (in the scenario considered, for $CW_{min} \approx 100$). This optimum value sits in the tradeoff between the decrease of the energy devoted to collisions and the increase in the energy spent when idling, and its computation is derived in Section 3.2.

Finally, we compare the efficiency η for three different WLAN scenarios (one for each of the interfaces of Table 1) and $N = 10$. We compare the numerical values given by simulations against the ones provided with our simplified analytical model, i.e., using (3) but substituting e with \hat{e} . We can see that the model is quite accurate, in particular in the relatively “flat” region where the efficiency is maximum, and that the optimal value of CW is different for each of the WLAN scenarios—a result we analyze next.

3 Configuration of 802.11

We provide in this section closed-form expressions for the optimal transmission probability τ , depending on the optimization objective throughput maximization in Section 3.1, and energy optimization in Section 3.2. Note that if we set

$CW_{min} = CW_{max}$, the transmission probability τ is easily related to the CW to use as follows

$$CW = \frac{2}{\tau} - 1$$

3.1 Throughput Maximization

When optimizing throughput, it is well known that CSMA/CA algorithms have an optimal transmission probability that depends on the network load, in terms of traffic generated and number of contending stations. For the case of saturated 802.11 WLANs, Bianchi [2] analytically derived the optimal transmission probability τ by maximizing the following expression for throughput

$$R = \frac{p_s L}{T_{slot}}$$

where T_{slot} is the average slot duration, given by

$$T_{slot} = (1 - \tau)^N T_e + (1 - (1 - \tau)^N) T_s$$

This optimization is done by deriving the above with respect to τ , and solving a second-grade equation resulting from the approximation $\tau \ll 1$. This results

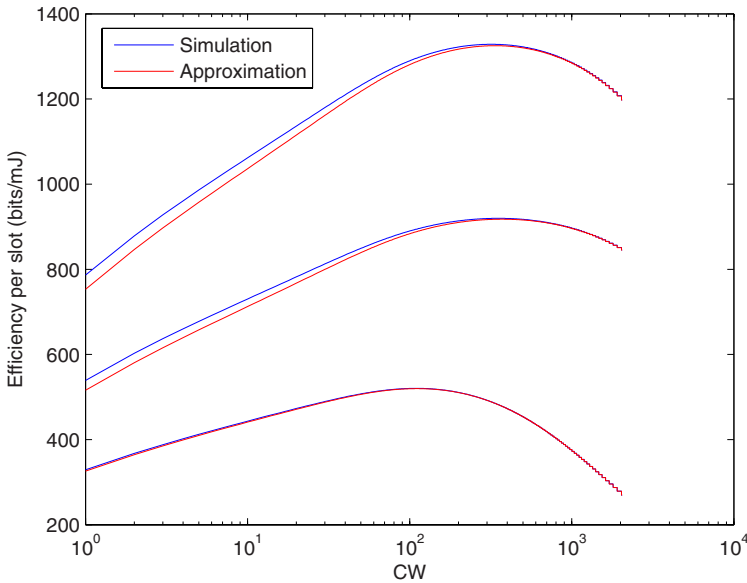


Fig. 3. Impact of the CW_{min} used on the efficiency. The arrays of curves from top to bottom show the results for energy profiles (A), (C), and (B).

in the following approximate value for the optimal transmission probability that maximizes throughput, τ_t

$$\tau_t \approx \frac{1}{N} \sqrt{\frac{2T_e}{T_s}} \tag{4}$$

Note that this optimal value of τ depends on the number of stations N , but also on the relative size of an empty timeslot T_e as compared to a timeslot that contains a transmission T_s . This way, apart from the number of stations, the ratio between the timeslot lengths sets the optimal tradeoff between the *cost* of a collision and the *cost* of idling. Indeed, this is the motivation behind some adaptive algorithms (e.g. Idle Sense [7]) that equalize the amount of time wasted in collisions with the amount of time waiting in backoff decrements.

However, because τ_t does not take into account energy consumption, for similar scenarios with different WLAN interfaces it will provide the same configuration for CW , while we have seen in Fig. 3 that the optimal CW value indeed depends on the energy consumption of the WLAN interfaces. This relationship is what we investigate in the next section.

3.2 Energy Optimization

To compute the transmission probability that optimizes the consumption of energy τ_e we start from the expression of η with the approximation for \hat{e}

$$\eta = \frac{\tau(1 - \tau)^{n-1}L}{R + \tau T' - (1 - \tau)^N R'}$$

And then compute the τ value that maximizes the above by

$$\frac{d\eta}{d\tau} = 0$$

This leads to the following

$$(n - 1)\tau^2 T' + (1 - \tau)^n R' + n\tau R - R = 0$$

By the following Taylor expansion of $(1 - \tau)^n$

$$(1 - \tau)^n \approx 1 - n\tau + \frac{1}{2}n(n - 1)\tau^2$$

We have the following equation

$$a\tau^2 + b\tau + c = 0$$

where

$$a = (n - 1)T' + \frac{1}{2}n(n - 1)R'$$

$$b = nE$$

$$c = -R$$

If we now define α and β as follows

$$\alpha = \frac{T'}{E} \quad , \quad \beta = \frac{R'}{E}$$

Then we have the following for the computation of τ_e :

$$\tau_e = \frac{-n + \sqrt{n^2 + 4(n-1)\alpha + 2n(n-1)\beta}}{2(n-1)\alpha + n(n-1)\beta}$$

That can be approximated as follows

$$\tau_e \approx \frac{1}{n} \sqrt{\frac{2}{\beta}} \approx \frac{1}{n} \sqrt{\frac{2\rho_e T_e}{\rho_r T_s}} \tag{5}$$

Note that, if we divide (4) by (5), we have that the relation between τ_t and τ_e is given by the ratio of the power consumption of the interface when receiving a frame over the power consumption when idling, i.e.,

$$\frac{\tau_e}{\tau_t} = \sqrt{\rho_r / \rho_e}$$

a relation that we analyze in the next section.

4 Energy Efficiency vs. Throughput Maximization

We first compare the resulting configuration obtained when maximizing throughput and when maximizing energy efficiency. To this end, in Fig. 4 we show the

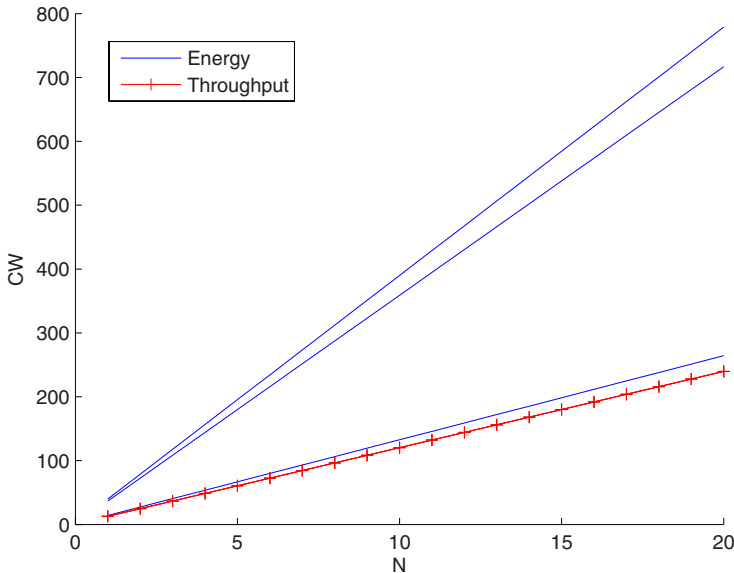


Fig. 4. Resulting CW configuration from each approach. The energy curves from top to bottom show the results for energy profile (C), (B), and (A).

resulting CW configuration for each maximization variable, for the three considered interfaces of Table I and an increasing number of stations N . From the figure is obvious to see that, while the throughput maximization provides the same CW for a given number of stations, the optimal CW for energy efficiency depends quite noticeably on the power characteristics of the WLAN interface. It can be seen that, the larger the ρ_r/ρ_e ratio, the larger the CW . This could be expected from the results of Fig. 5, as collisions have a larger cost and therefore it is more efficient to spend more time on the backoff, instead of taking the risk of transmitting and suffering from a no-success but energy-consuming collision.

We next compare the performance of both approaches, both in terms of energy efficiency and in terms of throughput, to gain further in the behavior of the WLAN under the different criteria. Results for each approach, as well as for the standard recommended values (DCF), are provided in Figs. 5 and 6, and can be summarized as follows:

- Considering energy efficiency, despite both throughput and energy optimizing approaches substantially outperform the DCF default configuration, the maximum efficiency approach provides the larger values of bits per Joule. As expected from the results of Fig. 4, the larger the ρ_r/ρ_e ratio, the larger the differences in performance between τ_e and τ_t .
- Considering throughput performance, it is clear that τ_t provides the largest values, as expected. It is quite remarkable, on the other hand, that while for one case the energy consumption provides almost the same results (this will

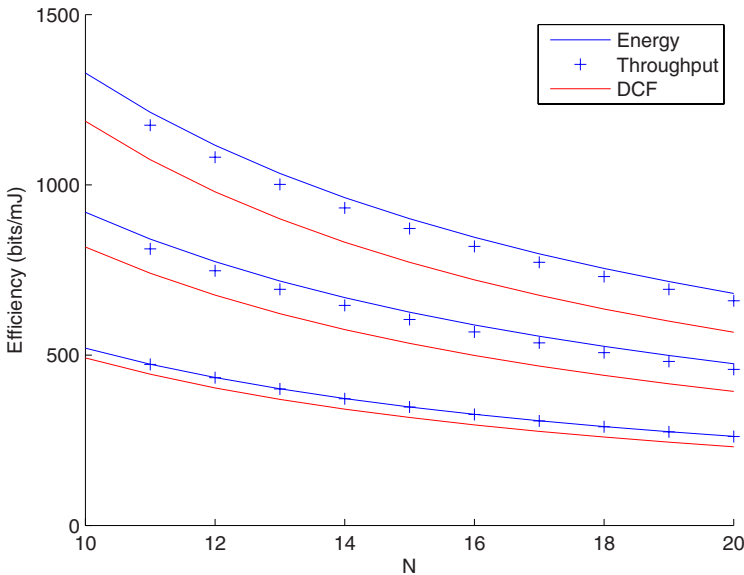


Fig. 5. Energy efficiency of each approach. The arrays of curves from top to bottom show the results for energy profile (B), (C), and (A).

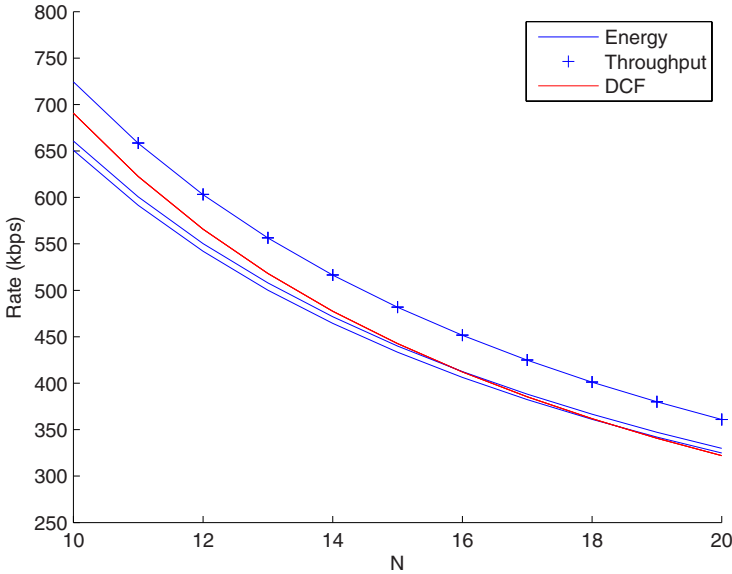


Fig. 6. Throughput performance of each approach. The energy curves from top to bottom show the results for energy profile (A), (C), and (B).

happen as long as $\sqrt{\rho_r/\rho_e} \approx 1$), for the other two cases there is a price to pay. Indeed, the throughput for these two interfaces is smaller than the one provided by DCF for $N \leq 17$. However, this slightly smaller throughput is obtained with a different CW value that results in quite different values of energy spent in collisions and backoff counter decrements.

Therefore, results confirm that there is a tradeoff between energy and throughput maximization, that depends on the characteristics of the WLAN interface. Indeed, for some ratios of power consumption we have the same result of [3], that both throughput and energy efficiency can be simultaneously maximized. However, our results show also that, for existing WLAN interfaces, this is not always the case, and there is a price to pay in throughput to achieve the most efficient behavior.

5 Conclusions

Greening the communication protocols is recognized as a primary design goal of future global network infrastructures. This paper presents a three-fold contribution on this field. First, it provides an approximate analytical model for the energy consumption of IEEE 802.11 LANs. Second, it defines an optimal configuration strategy that minimizes energy consumption for within such networks. Eventually, it provides a comparison of energy minimization against throughput

optimization, this way assessing the price to pay. Our future work will focus on experimental analysis and measurements on the field.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 214994. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express of the CARMEN project or the European Commission.

References

1. Baiamonte, V., Chiasserini, C.-F.: Saving energy during channel contention in 802.11 wlans. *Mob. Netw. Appl.* 11(2), 287–296 (2006)
2. Bianchi, G.: Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 18(3), 535–547 (2000)
3. Bruno, R., Conti, M., Gregori, E.: Optimization of efficiency and energy consumption in p-persistent csma-based wireless lans. *IEEE Transactions on Mobile Computing* 1(1), 10–31 (2002)
4. Chen, J.-C., Cheng, K.-W.: Edca/ca: Enhancement of ieee 802.11e edca by contention adaption for energy efficiency. *IEEE Transactions on Wireless Communications* 7(8), 2866–2870 (2008)
5. Ergen, M., Varaiya, P.: Decomposition of energy consumption in ieee 802.11. In: *IEEE International Conference on Communications, ICC 2007*, June 2007, pp. 403–408 (2007)
6. Feeney, L., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, vol. 3, pp. 1548–1557. IEEE, Los Alamitos (2001)
7. Heusse, M., Rousseau, F., Guillier, R., Duda, A.: Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. *SIGCOMM Comput. Commun. Rev.* 35(4), 121–132 (2005)
8. Wang, C., Li, B., Li, L.: A new collision resolution mechanism to enhance the performance of ieee 802.11 dcf. *IEEE Transactions on Vehicular Technology* 53(4), 1235–1246 (2004)
9. Wang, X., Yin, J., Agrawal, D.P.: Analysis and optimization of the energy efficiency in the 802.11 dcf. *Mob. Netw. Appl.* 11(2), 279–286 (2006)
10. Wu, H., Peng, Y., Long, K., Cheng, S., Ma, J.: Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement. In: *Proceedings of IEEE INFOCOM 2002 (June 2002)*

On the Effects of Transmit Power Control on the Energy Consumption of WiFi Network Cards

Francesco Ivan Di Piazza, Stefano Mangione, and Ilenia Tinnirello

Department of Electrical, Electronic and Telecommunication Engineering (DIEET),
Università di Palermo, 90128 Palermo, Italy

Abstract. Transmit power control has been largely proposed as a solution to improve the performance of packet radio systems in terms of increased throughput, spatial reuse and battery lifetime for mobile terminals. However, the benefits of transmit power control schemes on these different performance figures may strongly depend on the employed PHY technology and channel access mechanism. In this paper, we focus on the effects of power control on the energy consumption of WiFi network cards. By means of several experimental tests carried out under different operation conditions and modulation schemes, we try to justify why the reduction of the transmission power has a marginal effect on the overall energy consumption.

1 Introduction

Today, the de facto standard for wireless Internet access is the IEEE 802.11 [1] technology for Wireless Local Area Networks (WLAN), also known to the general public under the name WiFi [2]. WiFi connectivity is integrated by default in every modern portable computer, laptop and palmtop. WiFi networks for wireless Internet connectivity are available in most airports, university campuses, offices, homes, as well as in many restaurants and cafeterias. WiFi is extensively integrated in dedicated devices such as cameras, electric utilities or parking meters, and even exploited in quite specific applications such as control of garden hose sprinkles.

Due to the impressive proliferation of devices equipped with WiFi interfaces and to the limited battery power they rely on, reducing the energy consumption of WLAN interfaces is becoming a very important research issue. Indeed, several energy saving mechanisms, based on different approaches, have been recently explored in literature. Some of these mechanisms try to minimize the time intervals during which the WLAN transceiver is turned on, by means of periodic switching to a low-power doze state [3]. Although these solutions are very effective in reducing the energy consumption, they present two major drawbacks: i) they might not be applicable to ad-hoc networks, ii) they might severely degrade the quality of service in the network. The first problem arises because ad-hoc nodes have limited buffer capability. Therefore, packets destined to a doze node could be lost before the awakening of the node. The second problem arises because the alternating presence of sleeping nodes changes continuously the network topology

and connectivity level. In these conditions, some forms of coordination or synchronization among the nodes are required for avoiding routing problems and reducing the transport delays [4]. Moreover, common WiFi interfaces exhibit very slow transition times from a doze to an awake state, which prevent limiting the delays through high-rate switching.

Another approach to energy saving is based on the control of the transmit power. According to this approach, the transmitting node uses the minimum transmit power level that is required to communicate with the desired receiver. This mechanism reduces the power consumption of the sending node and limits the interference to other networks, thus improving both energy and bandwidth consumption. Although transmit power control (TPC) is not natively provided in WiFi networks, several research proposals [5,6,7] and emerging standards [8,9] have considered its implementation. In [6], the authors propose to extend the CTS and DATA frames in order to signal the minimum signal strength that is acceptable at the receiver and transmitter side. Similar RTS/CTS modifications are considered in [5], where a joint use of TPC and rate adaptation is proposed, so that the proper PHY rate as well as the best transmit power level can be adaptively selected. Most of these proposals [7] quantify the energy saving provided by TPC in WiFi networks via simulation. These results are based on power consumption models of WiFi interfaces, which are summarized into a set of power consumption values referring to different node states (namely, transmitting, receiving, idle and doze). Obviously, the performance evaluation of these schemes strongly depends on the setting of these values.

In this paper we deal with the problem of quantifying the energy saving that can be provided in WiFi networks by means of TPC. To this purpose, we experimentally characterize the power consumption of some commercial WiFi cards under different transmit power levels. Our methodology, similarly to the methodology described in [10,11], is able to provide: i) a direct measurement of instantaneous card consumptions, and ii) an indirect measurement of average (or per-packet) energy consumptions. Differently from previous results, we are able to rigorously control the transmit power and to compare the OFDM and DSSS modulations. Our conclusions show that little space is left to TPC for effectively reducing energy consumption of WiFi cards, due to the power consumed in idle states.

The rest of the paper is organized as follows. In section 2, we briefly review the 802.11 standard in order to define different card states corresponding to different power consumptions. In section 3, we describe our experiments, by illustrating our methodological approach and our measurement elaborations. In section 4, we try to provide a card sub-system decomposition, enlightening the fixed power consumption overheads. Finally, section 5 concludes the paper.

2 Energy Consumption in WiFi Cards

Regardless of the specific card implementation, we can expect that the energy consumption of WiFi cards depends both on physical layer (PHY) and medium

access control layer (MAC) operations. As far as concerns the PHY layer, in current 802.11a/b/g standards different modulations (e.g. DSSS and OFDM) and coding schemes are available for frame transmissions. Each scheme corresponds to a different activity interval required for transmitting or receiving a frame, which leads to different energy consumptions. Moreover, each scheme also exhibits a different processing complexity, which may cause further differences in the instantaneous power absorption. As far as concerns the MAC layer, the WiFi standard is based on a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, called Distributed Coordination Function (DCF). DCF has been designed for optimizing wireless medium utilization while maintaining the protocol simplicity. Therefore, it is based on some design choices which do not take into account energy consumption problems. For example, the use of an asynchronous access protocol is intrinsically inefficient for the reasons discussed in this section.

DCF operations can be summarized as follows. A station with a new frame to transmit has to monitor the channel state, until it is sensed idle for a period of time equal to a Distributed InterFrame Space (DIFS). If the channel is sensed busy before the DIFS expiration, the station has to add a further backoff delay before transmitting, in order to avoid a synchronization with the transmissions of other stations. The backoff interval is slotted for efficiency reasons and is doubled (up to a maximum value) at each consecutive failed transmission. Frame transmissions have to be explicitly acknowledged with ACK frames, because the CSMA/CA does not rely on the capability of the stations to detect a collision by hearing the channel. The ACK frames are immediately transmitted at the end of a frame reception, after a period of time called Short InterFrame Space (SIFS) shorter than a DIFS. If the transmitting station does not receive the ACK within a specified ACK_Timeout, it reschedules the packet transmission, according to the given backoff rules.

These access operations imply that a new frame transmission can start at any time instants on the channel and active stations have to continuously monitor the wireless medium in order to intercept incoming frames. As a consequence, a station spends a significant amount of time in monitoring the channel, regardless of the presence of incoming or outgoing traffic. Summarizing, during the activity intervals, a WiFi card can be in various operational states, which include:

- transmission, when the card is involved in the physical irradiation of an ongoing frame;
- reception/overhear, when the card is involved in demodulating a frame destined to itself or to another station;
- idle, when the card is monitoring the channel, ready to reveal channel busy signals, but no signal is present;
- doze, when the card radio transceiver is turned off.

Different operational states correspond to different power absorptions. Let W_{tx} , W_{rx} , W_{idle} and W_{doze} be the generic power absorbed, respectively, in transmission,

reception, idle and doze state. Regardless of the card implementation, we can expect that $W_{tx} \geq W_{rx} \geq W_{idle} \geq W_{doze}$.

Assuming that no power saving mechanism is employed (i.e. the card never switches to the doze state), the minimum energy $E_{min}(T)$ consumed in a given activity interval T is:

$$E_{min}(T) = W_{idle} \cdot T$$

This minimum consumption is experienced when the card does not transmit and receive any frames during the whole activity time. Conversely, the energy consumption is maximized when the card spends the maximum possible time in the transmission state. Since the standard limits the maximum frame size, this condition is verified when i) the card transmission buffer is never empty (i.e. the card works in saturation conditions), ii) the frames are transmitted at the minimum PHY rate, and iii) no other station accesses the channel. The ratio tx of the time spent in transmission for a card working in saturation conditions, in absence of contending stations, can be easily evaluated by considering the beginning of a new transmission as a regeneration instant. Specifically, being \bar{b} , T_{DATA} , and T_{ACK} , respectively, the average time spent in backoff, in transmitting a data frame and in receiving an ACK frame, it results:

$$tx = \frac{T_{DATA}}{T_{DATA} + SIFS + T_{ACK} + DIFS + \bar{b}} \quad (1)$$

For example, for the maximum admissible payload size of 2304 byte and the 802.11g PHY, it results $tx = 0.95\%$ at 6 Mbps and $tx = 0.70\%$ at 54 Mbps. The ratio rx of the time spent in reception corresponds to the ACK duration ratio within a transmission cycle, i.e.:

$$rx = \frac{T_{ACK}}{T_{DATA} + SIFS + T_{ACK} + DIFS + \bar{b}} \quad (2)$$

For example, for the previous case of 802.11g PHY with a payload length of 2304 byte and a data and acknowledgment rate of 6 Mbps, it results $rx = 1.2\%$. Given the tx ratio and rx ratio, the average power consumption \bar{W} can be evaluated as:

$$\bar{W} = tx \cdot W_{tx} + rx \cdot W_{rx} + (1 - tx - rx) \cdot W_{idle} \quad (3)$$

Therefore, the energy $E(T)$ consumed during T results:

$$E(T) = \bar{W} \cdot T \leq [txW_{tx} + (1 - tx)W_{idle}] \cdot T = E_{min} + tx \cdot (W_{tx} - W_{idle}) \cdot T$$

3 Energy Consumption Measurements

3.1 Methodology

To the best of our knowledge, in literature there are a few detailed measurement studies of the energy consumption of WiFi Cards. These studies can be divided into two general approaches: i) indirect measurements, obtained by monitoring

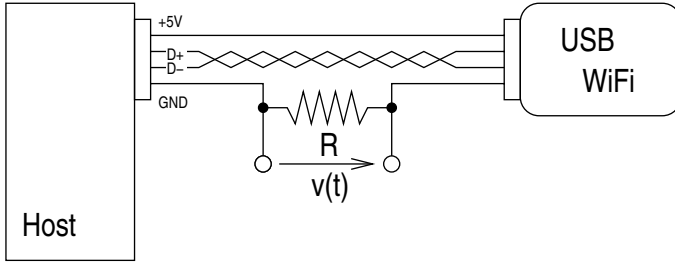


Fig. 1. Power measurement setup

the total energy consumed by laptops whose WiFi interface is enabled or disabled, ii) direct measurements, obtained by monitoring the input current drawn by the network card. We followed this second approach, for the case of USB WiFi cards. In fact, for these cards, it is immediate to probe the input current, by accessing the ground wire of the USB cable. Specifically, as shown in figure 1, we inserted a test resistor along the ground wire, in series with the card, and we measured the voltage at the resistor. Measurements were obtained using a 500 MHz Agilent digital oscilloscope, devised to acquire a complete voltage trace during an acquisition interval T . By opportunisticly tuning the temporal granularity of the oscilloscope traces, we are able to monitor the current values drawn during frame transmissions, frame receptions, channel monitoring and backoff. The instantaneous power consumptions are then evaluated, in the hypothesis of fixed input voltage $V_{in} = 5V$ and resistive input impedance of the card, as:

$$P(t) = V_{in} \frac{v(t)}{R}$$

where $v(t)$ is the direct measurement of the test resistor voltage, and $v(t)/R$ is the indirect measurement of the current drawn by the card. Elaborating the oscilloscope traces, we also averaged the instantaneous values for characterizing the W_{tx} , W_{rx} and W_{idle} values and the overall average consumption \overline{W} . In order to cross-validate our results, we performed some additional measurements by means of a digital multimeter. This instrument allows tracking the average power consumption at time scales much longer than a frame transmission time (e.g. 1 second). Thus, we compared these average values with the elaborations of the oscilloscope traces.

Although the results presented in this paper mainly refer to the D-Link DWL G-122 card, based on the Ralink chipset RT2500USB, we repeated our measurement campaign for other test cards (namely, Netgear WG111v2, Asus WL-167G and Linksys WUSB 300N), and for different operating systems (Windows and Linux). The host laptop was an Acer Extensa 5220, connected in ad-hoc mode with another identical laptop. As a traffic generator, we used the Iperf [12] tool with a CBR source over UDP. Unless otherwise specified, the source rate has been set to 100Mbps (in order to guarantee saturation of the transmission buffer)

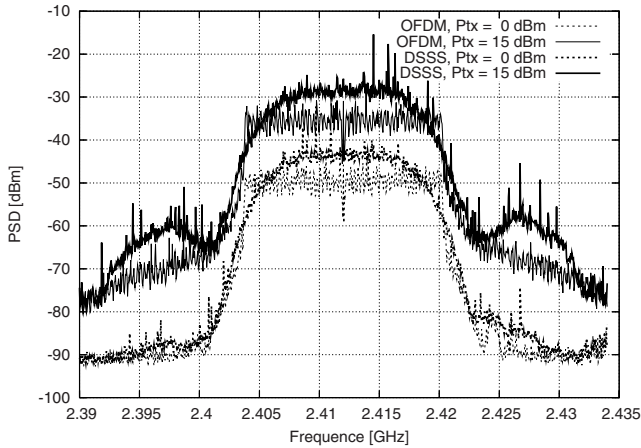


Fig. 2. Power Spectral Density of OFDM and DSSS signals, for $P_{tx} = 15$ dBm and $P_{tx} = 0$ dBm

with a frame length equal to 1470 bytes. We ran different experiments, changing the PHY transmit rate r and the PHY transmit power P_{tx} employed by the cards. These parameters have been changed by means of the card configuration interface at the driver level. In some cases (e.g. the very recent Linksys card), some configuration options were not available. Therefore, we used the D-Link card as a reference card thanks to the availability of a full featured driver.

We carefully checked that the values specified at the driver level were conform to the actual values adopted by the cards. About the PHY transmit rate, we considered a very simple validation test, by comparing the actual frame transmission times with the expected ones. The actual frame transmission times have been measured at the oscilloscope, by identifying time intervals during which the card drew the maximum current value. About the PHY transmit power, we monitored the RSSI values sampled at the receiver for different configuration of the transmit power, while maintaining the transmitter and the receiver node at the same position. We noticed that the RSSI values experienced increments or decrements corresponding exactly to the changes applied at the transmitter side. Some exceptions have been found when we set transmit power values higher than 15 dBm. In fact, despite the regulatory limit is higher, some cards do not allow settings higher than 15 dBm. Finally, we also checked that the power spectral density (PSD) revealed by means of a spectrum analyzer changed in agreement with the PHY transmit power. Figure 2 plots some traces of our spectrum analyzer, obtained for $P_{tx}=15$ dBm and $P_{tx}=0$ dBm, in the case of $r=6$ Mbps (OFDM modulation) and $r=11$ Mbps (DSSS modulation).

3.2 Impact of Transmit Power

Figures 3 and 4 plot the power absorption traces collected during some experiments lasting $T=5$ ms. The figures refer to the D-Link DWL G-122 card and

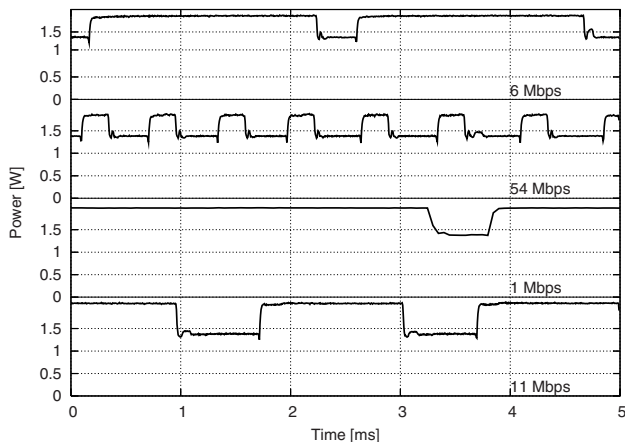


Fig. 3. Instantaneous power consumption in saturation conditions for different transmit rates - $P_{tx} = 15$ dBm

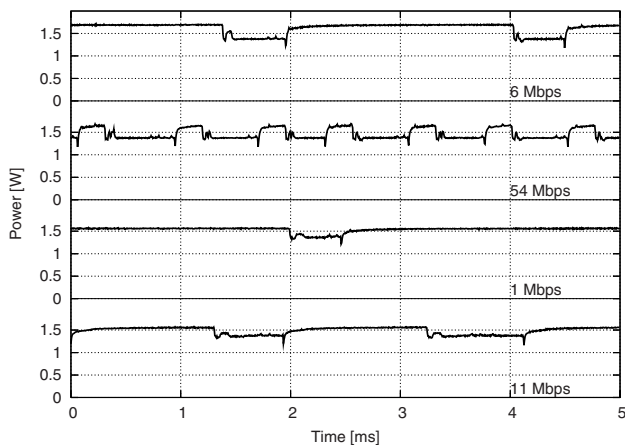


Fig. 4. Instantaneous power consumption in saturation conditions for different transmit rates - $P_{tx} = 0$ dBm

have been obtained for $P_{tx}=15$ dBm (figure 3) and $P_{tx}=0$ dBm (figure 4) at different transmit rates (namely, 1 Mbps, 6 Mbps, 11 Mbps and 54 Mbps). Unless otherwise specified, we always refer to this test card.

Focusing on figure 3, we can easily recognize the different working states of the card under test. The higher power levels correspond to the transmission states, whose duration depends on the employed rate. The time intervals between two consecutive transmissions correspond to the reception of the ACK frames and to the subsequent random backoff process. The figure visualizes that the power consumption experienced during these two phases, i.e. in reception and idle mode, is

substantially the same. In order to better visualize the ACK reception times, we set the network basic rate at 2 Mbps. In each trace, we can recognize a narrow spike over the lower level at the end of each frame transmission, which corresponds to the ACK reception. For the 54 Mbps trace, we can observe two small spikes between the transmission of the sixth and seventh frame. We verified, by means of a traffic sniffer, that this spike is due to the reception of a beacon frame transmitted by the receiver¹.

By comparing figure 3 and figure 4, it is qualitatively evident that for $P_{tx}=0$ dBm the power W_{tx} consumed in the transmission state is reduced. However, such a reduction is marginal for the OFDM modulated frames (i.e. for the 6 Mbps and 54 Mbps cases), while is appreciable for the DSSS ones. The power consumption experienced in reception and idle state is approximately the same in both the figures.

Table 1 quantifies our previous considerations. We estimated the W_{tx} , W_{rx} and W_{idle} values, by quantizing the traces plotted in figures 3 and 4 into three different levels (an high level for the transmission state, an intermediate level for the reception state, and a low level for the idle state), and by averaging the instantaneous values collected for each level. By using these estimates, we evaluated the average power consumption according to 3 and we compared such an evaluation with the trace average values and with the multimeter measurements. The average values have been summarized under the \overline{W} column and identified, respectively, by the *Eqn*, *Osc* and *Mul* label. The results obtained with the three different methodologies are in good agreement. Since equation 3 is based on the computation of the frame transmission times, the agreement of these results also proves that the actual transmission rate is equal to the nominal one, set at the driver level.

From the table, we can observe that the power consumed in reception (W_{rx}) and idle (W_{idle}) state are comparable in all the cases. By reducing the transmit power P_{tx} from 15 dBm to 0 dBm, the W_{tx} values are reduced of about 20% for $r=1$ Mbps and $r=11$ Mbps (DSSS case), and about 10% for $r=6$ Mbps and $r=54$ Mbps (OFDM case). These reductions are reflected in lower percentual reduction of the average power consumption \overline{W} . Note that the table refers to a card working in saturation conditions. Since in most cases the transmission time is a small fraction of the whole activity time, the reduction of the W_{tx} values by means of TPC has a marginal effect on the overall energy consumption of the cards.

Finally, table 2 summarizes the results of similar measurements carried out with different cards. From the table we note that, for each card, the W_{idle} and W_{rx} values are comparable. For the cards transmitting at 15 dBm, we also note that the power W_{tx} consumed in the transmission state may vary from 1.85 W up to 2.69 W because of different card designs and implementations.

¹ We recall that in ad-hoc networks, all the nodes schedule the beacon transmission at regular time instants. When a given node succeeds in transmitting the beacon, all the other pending ones are suspended.

Table 1. Per-state and average power consumption values [W]

r	W_{tx}		W_{rx}		W_{idle}		\overline{W}_{15dBm}			\overline{W}_{0dBm}		
	15 dBm	0 dBm	15 dBm	0 dBm	15 dBm	0 dBm	Eqn	Osc	Mul	Eqn	Osc	Mul
1 Mbps	1.98	1.54	1.40	1.40	1.38	1.38	1.94	1.94	1.96	1.52	1.49	1.53
11 Mbps	2.06	1.56	1.40	1.40	1.38	1.38	1.84	1.86	1.79	1.50	1.54	1.49
6 Mbps	1.85	1.64	1.44	1.44	1.38	1.38	1.77	1.77	1.74	1.60	1.62	1.59
54 Mbps	1.85	1.64	1.44	1.44	1.38	1.38	1.57	1.55	1.51	1.49	1.46	1.44

Table 2. Power consumption values for different cards for $r = 6$ Mbps [W]

Card	Ptx	W_{tx}	W_{rx}	W_{idle}
Linksys	15	2.69	1.65	1.61
Netgear	15	2.01	1.58	1.39
Asus	12	1.40	1.01	0.97
D-Link	15	1.85	1.44	1.38

Table 3. Average power [W], average throughput [Mbps], and energy per-bit [J/b] at different rates

r	\overline{W}	Thr	E(T)/bit
1 Mbps	1.94	0.915	2.12e-6
11 Mbps	1.86	6.192	3.00e-7
6 Mbps	1.77	4.458	3.97e-7
54 Mbps	1.55	13.706	1.13e-7

3.3 Impact of Transmit Rate

The most evident effect of the PHY transmit rate on energy consumption is obviously related to the duration of frame transmissions. As the transmit rate increases, the ratio tx spent by the card in transmission state is reduced, thus resulting in a lower average \overline{W} value. Moreover, the reduction of the transmission times allows to deliver an higher number of frames during T . Therefore, the per-bit energy consumption is further improved. Table 3 quantifies these considerations by summarizing the \overline{W} (which is proportional to the energy consumption $E(T)$), the average throughput, and the per-bit energy consumption observed in saturation conditions at different rates. From the table, we can conclude that the PHY transmit rate strongly affects the per-bit energy consumption of the cards.

In section 2, we have implicitly assumed that each card is characterized by a fixed W_{tx} value, which does not depend on the transmit rate, and that such a value is constant during the whole transmission interval T_{DATA} . However, these assumptions are not rigorous. In table 1 we can see a clear difference between the OFDM and DSSS modulations (W_{tx} is about 1.8 W for the OFDM case and about 2 W for the DSSS one). While in OFDM mode the W_{tx} is about the

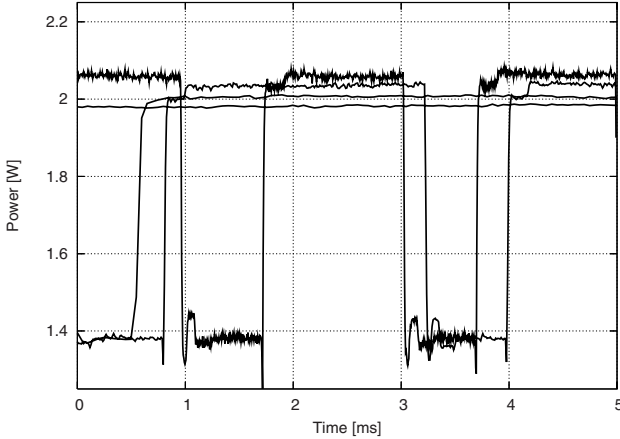


Fig. 5. Instantaneous power consumption at 1, 2, 5.5. and 11 Mbps

same for the 6 Mbps and 54 Mbps case, some differences appear in DSSS mode, as the transmit rate changes from 1 to 11 Mbps. In order to better visualize this phenomenon, figure 5 plots the instantaneous power consumption observed for the DSSS modulations. The traces collected at different rates have not been labeled, since we can easily recognize the 1, 2, 5.5 and 11 Mbps traces according to frame transmission duration.

From the figure it is evident that the instantaneous W_{tx} values slightly grow as the transmit rate increases. We suspect that this increment is due to the additional processing complexity introduced by the higher rate modulations. At the beginning of the frame transmissions, for the 5.5 Mbps and 11 Mbps traces, we can also recognize that the preamble transmission is characterized by a power consumption lower than during the rest of the frame.

4 Energy Consumption Components

The power consumption measurements described in the previous section have been obtained by considering the card under test (i.e. a D-Link DWL G-122 card) as a black box. In other words, we characterized the instantaneous power consumption without identifying the different hardware components responsible of partial absorptions. Indeed, the decomposition of the overall consumption into independent sub-systems performance can be very enlightening for the design of effective power saving schemes.

Figure 6 shows a card block diagram, analogous to the one depicted in 13. The card has been decomposed into: a Power Amplifier (PA), an RF subsystem (RF), a MAC/BaseBand processor, and a USB host interface (USB).

Each of these sub-blocks gives a different and easily recognizable contribution to power consumption. The Power Amplifier is relevant only during transmission

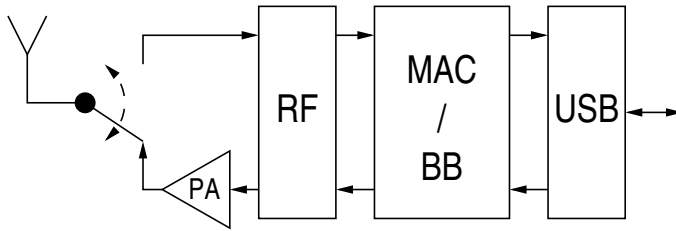


Fig. 6. System blocks of a USB WiFi card

bursts. Most WiFi implementations feature an external power amplifier. The reason for choosing an external power amplifier is that the realization of low-voltage CMOS linear power amplifiers for OFDM signals is an extremely challenging task. In fact, the OFDM signal has a very high Peak-to-Average Power Ratio (PAPR) which makes difficult designing efficient linear power amplifiers². The RF subsystem, which is responsible for frequency synthesis, synchronization, up and down conversion and low-noise amplification, absorbs power while the card is not dozen. The power consumption due to baseband processing is very different depending on if the station is transmitting or receiving. When the card is in transmission state, the baseband processor just encodes and modulates the frames, thus resulting in a very lower power consumption. Conversely, when the card is in reception state, several actions are needed, such as timing and fine frequency synchronization, channel estimation and equalization and, in the case of OFDM signals, channel decoding. All these operations make the baseband processing more power-eager during reception than during transmission. Since the MAC processing has an event-based low-rate schedule, its power consumption is very low. Finally, a component which turns out to have a significant contribution to the overall power consumption is the Universal Serial Bus interface to the host. In the following, we try to dissect separately the contribution of each component.

4.1 Power Amplifier

We can identify the power consumption W_{PA} due to the power amplifier by considering $W_{PA} = W_{tx} - W_{idle}$. From table [1](#), for a nominal Ptx value of 15 dBm, it results $W_{PA} \simeq 600$ mW in the case of DSSS modulations, and $W_{PA} \simeq 470$ mW in the case of OFDM modulations. Such values are compatible to a power amplifier efficiency of about 5%.

Note that the lower W_{PA} value experienced under the OFDM mode is not due to an higher efficiency in amplifying OFDM signals. In fact, by integrating the PSD traces collected by the spectrum analyzer, we found that, despite of

² The most efficient power amplifiers found in the literature have an efficiency which may approximately vary from 40% [\[14\]](#) down to less than 10% [\[15\]](#) as the amplifier gain increases.

the same nominal transmit power, the power radiated in OFDM mode is 4.4 dB lower than the power radiated in DSSS mode. This phenomenon can be explained as a side-effect of the non-linearity of the power amplifier. When operating in DSSS mode (i.e. with low PAPR), the power amplifier can be fed with high level signals, without triggering spectral spurs. Conversely, when operating on OFDM signals (i.e. with high PAPR), the signal levels have to be attenuated in order to avoid spur signals impairing the spectral mask requirements [2].

4.2 RF Front-End and Baseband Processing

We assume that the baseband power consumed when the card is in transmission state is negligible. As far as concern the reception state, we identify the power consumption W_{BB} due to the baseband processing as $W_{BB} = W_{rx} - W_{idle}$. From table I, it results $W_{BB} \simeq 20$ mW in the case of DSSS modulations, and $W_{BB} \simeq 60$ mW in the case of OFDM modulation. As expected, the W_{BB} computation leads to the same results in case of $P_{tx} = 15$ dBm and $P_{tx} = 0$ dBm.

In order to compute the RF front-end power consumption W_{RF} , we also measured the instantaneous power W_{doze} absorbed by our card while in doze state. The measurement has been carried out by switching the card transceiver off. By processing the oscilloscope traces, we obtained an average W_{doze} value of 760 mW. Assuming that W_{RF} is independent from the transmission or reception state, we consider $W_{RF} = W_{idle} - W_{doze} \simeq 620$ mW.

4.3 Universal Serial Bus/Host Interface

We assume that the power consumption resulting in the doze state is mainly due to the USB interface. Therefore, $W_{USB} \simeq W_{doze} = 760$ mW. Our measurements are compatible to the power consumption of a common USB / Host interface [13], which is about 600/700 mW. Note that this contribution represents an high fraction of the whole card consumption, being comparable to the W_{PA} value measured at full transmit power. This high value may be explained with the high speed of the PHY featured in the Universal Serial Bus specification [16].

5 Conclusions

In this paper we analyzed the power consumption of common USB WiFi cards, under different operation conditions. Specifically, we monitored the current drawn by the cards, focusing on a D-Link DWL G-122 card, for different PHY transmit rates and transmit powers. We found that reducing the transmit power has a little impact on the average energy consumption of the cards. This result, confirmed also in previous experiments [10], depends on the high power level absorbed when the card is idle, which represents a very high fixed overhead. We also found that transmit power control has a lower impact when the card works in OFDM mode rather than in DSSS mode. Finally, we tried to dissect our power consumption measurements, by identifying the consumption quota of different card subsystems, including the power amplifier, the RF-front end, the baseband and the host interface.

References

1. IEEE Standard 802.11 - 1999; Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (November 1999)
2. Wi-Fi Alliance, <http://www.wi-fi.org>
3. Simunic, T., Benini, L., Glynn, P., De Micheli, G.: Dynamic Power Management for Portable Systems. In: Proc. ACM MobiCom 2000, Boston, MA, August 2000, pp. 11–19 (2000)
4. Yong, H., Ruixi, Y.: A Novel Scheduled Power Saving Mechanism for 802.11 Wireless LANs. *IEEE Transactions on Mobile Computing* 8(10), 1368–1383 (2009)
5. Qiao, D., Choi, S., Jain, A., Shin, K.G.: Adaptive transmit power control in IEEE 802.11a wireless LANs. In: Proc. IEEE VTC 2003, April 2003, vol. 1, pp. 433–437 (2003)
6. Agarwal, S., Katz, R.H., Krishnamurthy, S.V., Dao, S.K.: Distributed power control in ad-hoc wireless networks. In: Proc. IEEE PIMRC 2001, September 2001, vol. 2, pp. 59–66 (2001)
7. Ebert, J.P., Wolisz, A.: Combined tuning of RF power and medium access control for WLANs. *Mobile Networks and Applications* 5(6), 417–426 (2001)
8. Qiao, D., Choi, S.: New 802.11h mechanisms can reduce power consumption. *IEEE IT Professional* 8(2), 43–48 (2006)
9. Kongseng, A., Hossain, Z., Gorg, C.: Transmit Power Control Algorithms in IEEE 802.11h Based Networks. In: Proc. IEEE PIMRC 2005, September 2005, vol. 3, pp. 1441–1445 (2005)
10. Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proc. IEEE INFOCOM 2001, April 2001, vol. 3, pp. 1548–1557 (2001)
11. Ebert, J.P., Burns, B., Wolisz, A.: A trace-based approach for determining the energy consumption of a WLAN network interface. In: European Wireless 2002, February 2002, pp. 230–236 (2002)
12. <http://sourceforge.net/projects/iperf>
13. Keng Fong, D., Tung, M., Lee, S., Lee, B., Wu, P., Cheng, T., Pare, J., Feng, J., Chang, E., Simpson, J., Wong, F., Jann, K., Liao, D.: A Complete Dual-band Chip-set with USB 2.0 Interface for IEEE 802.11 a/b/g WLAN applications. In: Asian Solid-State Circuits Conference 2005, November 2005, pp. 257–260 (2005)
14. Huang, C.-C., Chen, W.-T., Chen, K.-Y.: High Efficiency Linear Power Amplifier for IEEE 802.11g WLAN Applications. *IEEE Microwave and Wireless Components Letters* 16(9), 508–510 (2006)
15. Wang, P.-C., Huang, K.-Y., Kuo, Y.-F., Huang, M.-C., Lu, C.-H., Chen, T.-M., Chang, C.-J., Chan, K.-U., Yeh, T.-H., Wang, W.-S., Lin, Y.-H., Lee, C.-C.: A 2.4-GHz +25dBm P-1dB linear power amplifier with dynamic bias control in a 65-nm CMOS process. In: 34th European Solid-State Circuits Conference, September 2008, pp. 490–493 (2008)
16. Universal Serial Bus Revision 2.0 specification, <http://www.usb.org/developers/>

A Novel Power-Efficient Middleware Scheme for Sensor Grid Applications

Nikolaos I. Miridakis, Vasileios Giotsas, Dimitrios D. Vergados,
and Christos Douligeris

Department of Informatics University of Piraeus 80, Karaoli & Dimitriou St.,
GR-185 34 Piraeus, Greece
{nikozm, vergados, cdoulig}@unipi.gr, v.giotsas@ieee.org

Abstract. Sensor grid deployments integrate wireless sensor networks (WSNs) and Grid Computing (GC) into a merged platform. A middleware architecture is a prerequisite for sensor grids in order to bridge the two heterogeneous technologies and efficiently support aggregated grid services available to a large number of grid users. On the other hand, the energy conservation of the participating sensor nodes is an essential factor for QoS provisioning, thereby extending WSNs survivability and providing diversity to potential grid services. For the best of our knowledge, power awareness for middleware architectures for sensor grids has never been studied in the literature so far. The rationale of our work employs a scheduler which provides QoS to the grid users from an energy awareness perspective by interacting with an appropriate resource manager. Our simulations show the effectiveness of the proposed scheme whereas a proxy-based middleware for sensor grids has been adapted.

Keywords: Sensor Grids, Power Efficiency, QoS scheduling.

1 Introduction

Wireless sensor networks (WSNs) are one of the most rapidly evolving research and development fields for microelectronics. Their applications are countless, and the market potential is huge. Recent advances in micro-electromechanical systems (MEMS) have led to the creation of small sensor nodes which integrate several kinds of sensor components such as a central processing unit, memory and a wireless transceiver [1, 2]. These sensor components have been characterized as low-cost, low-power and self-contained instruments with limited sensing, data processing, and wireless communication capabilities. The most important applications of WSNs include environmental and habitat monitoring, healthcare monitoring of patients, weather monitoring and forecasting, military and homeland security surveillance, tracking of goods and manufacturing processes and safety monitoring of physical structures and construction sites, smart homes and offices [3].

Nevertheless, sensor nodes still remain resource constrained due to their limited bandwidth range and computation capabilities. Thankfully, WSNs consist of hundreds (sometimes thousands) sensor nodes deployed and aggregated over a certain wide

area, so the computational burden is therefore distributed among the nodes. Thus, WSNs are important distributed computing resources that can be shared by different users and applications [4].

Grid Computing provides a federation of heterogeneous computational servers and collaborating systems which are communicating through high-speed network connections. Many industries have recognised the importance of grid computing for 'e-science' where the grid has been employed extensively in the fields of bioinformatics, engineering design, business, manufacturing, environmental control and weather forecasting [1, 5, 6].

The combination of WSNs and grid computing under a sensor grid architecture (*sensor grid* in short) takes advantage of all the strengths and benefits of sensor networks and grid computing resulting in a single integrated platform [1, 7]. Thus, a sensor grid may combine real-time data about a wide unit area with vast computational resources derived from the grid architecture. A typical sensor grid framework is shown in Figure 1a. There is a trade off thought between the two merged technologies. On one hand, sensor nodes have constrained resources as they monitor the environment on a real-time basis while the resource-full grid infrastructure promises solutions to computational and communication tasks according to the ever-increasing needs of users.

There are mainly two approaches on a sensor grid deployment; the centralized and the distributed approach. In the centralized, sensor nodes and sensor networks are connected directly to the grid. High-speed communication links are necessary for this approach where all computational tasks take place on the grid. The main drawback of this approach is the fact that it leads to excessive communications among the nodes which rapidly depletes the batteries resulting to network partitioning, a rather undesirable choice. Also, possible communication failures in some nodes, such as bad radio propagation conditions, jamming and interference, could result in a general breakdown of the system. The distributed approach is more robust and efficient technique since it allows all computational and decision making jobs to be performed within the sensor network according to their resources and capabilities at a real-time basis [1].

Sensor grids being a relatively new area of research, there are many issues left unaddressed regarding their design. Moreover, because WSNs are usually based on proprietary designs and protocols, it is a challenging task to integrate them with the standard grid architecture and protocols [3]. In this paper, we analyze the issues and challenges present in the integration of WSNs and the Grid considering a distributed approach managed locally in each sensor area (a sensor area consists of a WSN or a cluster of several WSNs connected with the grid via virtual organizations). We also describe a proxy-based architecture, a middleware component enhancing the effectiveness of the overall framework, as shown in Figure 1b.

The middleware plays the role of an appropriate interface which is capable of providing functionalities such as normalizing and synchronizing the communication between sensor nodes and the grid. Furthermore, the proxy middleware model takes into consideration the limited power resources of sensor nodes making it an energy-aware architecture which has as its main scope the preservation of the resources of

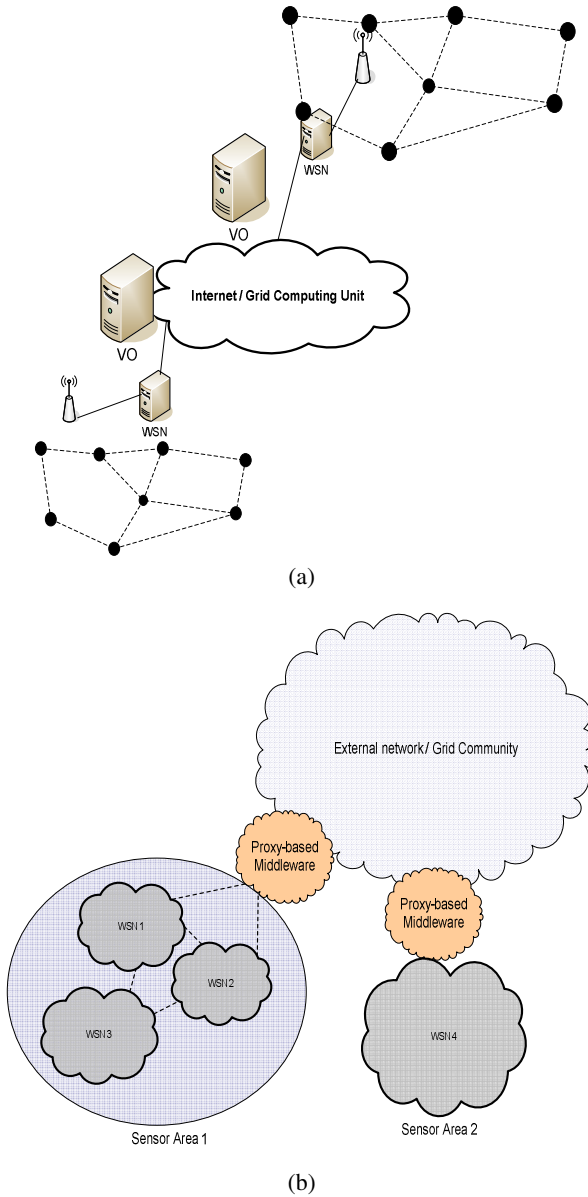


Fig. 1. (a) Sensor Grid framework. (b) The middleware connecting platform.

individual wireless nodes. As many symbols are used in this paper, Table 1 summarizes the most important ones.

Table 1. Summary of important symbols used

Symbol	Definition
d_{ij}	Distance between node i to j
d_{eff}	Maximum effective transmission range
V_g	User group credentials
V_u	User profile
C	Service Class
N_S^c	Number of services for service class C
\hat{P}^j	Power consumption of node j for a single service
P_c^j	Total power consumption of node j for service class C
$Sh_P_x[]$	An array which holds all shortest paths within subset X
m	Number of hops from a requested node to the gateway
ξ	Number of all paths from a node to the gateway
$sig_P_c^j$	Signaling cost from the gateway to node j

The rest of this paper is organized as follows. In section 2, we discuss the most important compatibility problems that the two technologies encounter in order to provide an integrated platform. In section 3, the proposed model is presented in detail. Performance evaluation results are given in Section 4, followed by concluding remarks in Section 5.

2 Design Issues and Challenges

In this section we discuss the most important differences of the two merged technologies within sensor grid architecture. A natural approach to integrate sensor nodes into the grid is to adopt the grid standards and APIs. The Open Grid Services Infrastructure (OGSI) [14] establishes web services based on XML, SOAP and WSDL formats. However, since sensor nodes have limited resources and computational capabilities, as mentioned earlier, it may not be feasible to manipulate sensor data and to encode them into SOAP envelopes using XML formats. Therefore, many grid services may be too complicated for the capabilities of the common wireless sensor nodes [6].

Moreover, most grid processes and applications use existing internet protocols to exchange messages, e.g. TCP, FTP, HTTP. Sensor networks, on the other hand, make use of low-level protocols (energy efficient MAC and routing protocols) due to their nature [2, 13]. Hence, the direct communication of a WSN with the grid is not feasible without appropriate interface.

Power management is one of the major issues in WSNs and sensor grid deployments. The grid infrastructure must be aware of the power/energy status of the nodes of each sensor area in order to make the most efficient and robust decisions

since the availability of a WSN depends not only on its average load but also on its power resource constrains [3, 8].

An appropriate scheduler is definitely one of the most important and most complex operations in a proxy-based middleware deployment for sensor grids. The role of a scheduler in a typical WSN is to achieve load balancing and to avoid energy dissipation among the nodes, thus extending the network's lifetime and preventing network partitioning. In a sensor grid infrastructure, sensor nodes might be consisting of several wireless sensors, each used for a different purpose (temperature, sound, light, vibration etc) [3, 5]. Each type of data is collected by different types of wireless transceivers placed on the same device. A scheduler (combined with an appropriate resource manager, as explained in the next section) should control the on/off mode of the transceivers of every sensor device within the sensor area according to the needs of grid users as well as the available sensor resources. Furthermore, quality of service (QoS) is one very important issue in sensor grid networks. QoS is associated with the personalization logic that defines the service class for each user or user group responding to the grid. The personalization logic within the grid infrastructure branches the available services into classes according to the type of user, e.g. administrator, unsubscribed user, academic staff, commercial user, government. QoS factor is performed along with personalization logic coherently. Thus, an efficient scheduler should take into account the personalization logic, the QoS and the resource constraints of the nodes in a sensor area in order to provide suitable services both for the WSN and the grid.

3 Description of the Proxy-Based Middleware

In this section we describe the proposed middleware framework and we analyze its components and their functionalities in detail. The diagram in Figure 2 shows the deployment of the proposed proxy-based middleware infrastructure [9].

3.1 System Overview

The gateway is a station which collects the information from all the sensor nodes within its sensor area through the reception of wireless MAC frames transmitted directly from the WSNs. It also holds a record file with all the sensor ids which participate in the communication process (to be discussed subsequently). Afterwards, the gateway sends the aggregated row data flows to the middleware via a wired link. First, the preprocessing component evaluates the received data by purifying them from possible anomalies and aberrations due to propagation attenuation and atmospheric interference that the transferring process might cause. It also isolates individual MAC frames from the consecutive data flow and passes them to the filtering component. Then, an extraction and classification regarding the content of the frames takes place according to the *[source/destination]* value. Subsequently, an XML message conversion follows in order to achieve the appropriate compatibility with the corresponding grid applications and requests. The XML converter imports the sensor data to XML files according to predefined formats derived from the available XML database.

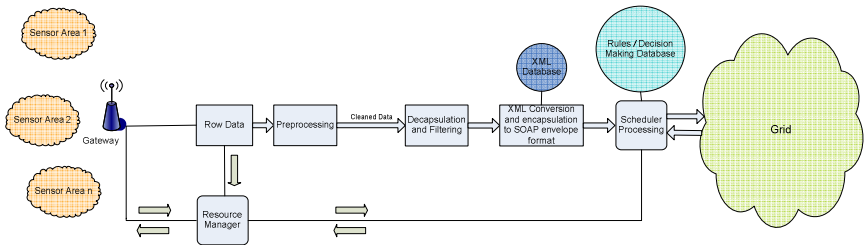


Fig. 2. Data flow in the proposed Sensor Grid framework

The XML output file depends on both the service/user class and the [source/destination] value. Hence, the final filtration of the overall message is accomplished at the scheduler module where the decision for the latter file format depends on the information by the QoS database where all rules for the grid services and the users take place. Along with the QoS database, the scheduler is directly communicating with the resource manager module which is also responsible for service class classification. Furthermore, the resource manager sends statistics for the average power utilization that has been observed in each sensor area and WSN to the scheduler in fixed time intervals. This information is derived by the MAC protocol that is used in each sensor area where an appropriate traffic monitoring of each sensor device (or sensor id) is achieved. Based on the aggregated resources that have been consumed in each sensor area, a classification of the available QoS is done and access is given or denied to grid applications according to their specific requirements.

The preprocessing and filtering components of the above mentioned framework implement PHY layer functionalities, as mentioned earlier. Therefore, the modeling of these components represents a task which is out of the scope of this paper. Our main interest has been placed on the modeling of an appropriate scheduler and a resource manager in order to provide energy awareness to sensor grids deployments as figure 3 shows.

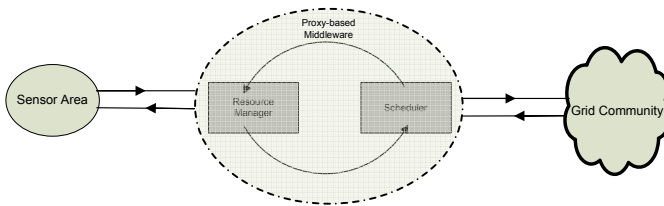


Fig. 3. Interaction of the main middleware components

3.2 Sensor Area

The sensor area represents the target interest of grid users in a sensor grid infrastructure. It contains a certain number (1 to N) of different WSNs; each WSN implements its MAC processes independently in order to serve grid calls. All WSNs in a sensor area communicate with the same gateway. The gateway is responsible for

the coordination of communication within the WSN by sending signaling messages to all the nodes that participate in each grid call. In particular, upon a grid call request, the gateway notifies the requested node to measure the requested data information and all the intermediate nodes to the path from that node to the gateway. This latter procedure is accomplished via signaling messages which hold the included sensor ids after the instructed decision of the resource manager (as discussed later on). You may notice that each WSN in a sensor area could implement different MAC protocols in a fully distributed manner as long as the communication is accomplished via the same gateway. The proposed proxy-based middleware defines the routing strategy that should be followed, through an appropriate resource manager from a network perspective.

For the modeling of the WSN we consider a directed acyclic graph DAG (G, E) where all vertices correspond to the nodes of the network. The terms vertex and node will be used interchangeably in the sequel. An edge E between two vertices i and j , exist iff (if and only if) node i is adjacent to node j and can communicate directly with it. Thus, there is an edge $E_{i,j}$ iff $d_{i,j} \leq d_{\text{eff}}$, where $d_{i,j}$ denotes the distance between nodes i and j and d_{eff} denotes the effective maximum distance due to the transmission range of the wireless sensors. We also assume, without loss of generality, that the transmission range is the same for all the sensor nodes. The direction of the data flow is always from the polled node (transmitter) to the gateway (receiver) of the considered graph. Prior to the message exchanging procedure, the signaling mode takes place. The signaling mode is modeled by a graph DAG' (G', E') with the opposite direction (from the gateway to the requested node and all intermediate nodes that will participate in the communication process).

Note that the above mentioned routing decision is accomplished in the network layer perspective. To provide multi-hop relay services in a WSN, or a sensor area in general, the resource management at the link layer and the routing at the network layer interact with each other. As the first step in our research, we consider separate designs at the resource management on routing and the resource management on MAC, and assume that a MAC protocol is already in place. How to achieve an optimal or suboptimal joint design of routing and resource management is very important issue for further research.

3.3 Scheduler

The scheduler module provides the interface of the middleware with the external grid community. Upon a grid call, the scheduler verifies the request id according to specific validation criteria which are stored in a user profile database. The authorization and the authentication of incoming grid calls are accomplished by consecutive interactions of the scheduler with the database. Each user has to register to the system in order to get access to the sensor areas. According to registered profiles the scheduler decides for the service class that could be supported from the system hereafter. If the requested service class matches the credentials of the associated user profile the procedure continues, otherwise there is a drop call event due to lack of necessary credentials.

More specifically, the system supports a fixed number of user groups, each with different access rights. User group entries can be defined as:

$$V_g = [\text{res_util}, \text{group_thr}], \quad g = 1, 2, 3, \dots, N \quad (1)$$

where V_g is a vector which contains the credentials for the specific user group g , res_util denotes the maximum resource utilization percentage per request per node for the current user group and group_thr denotes the maximum resource utilization per request per node when the available resources of a considered node are equal to the minimum allowed energy threshold of sensor node upon the current request.

Every user has a registered entry in one of the above mentioned user group profiles which is stored in the database in the following format:

$$V_u = [\text{user_id}, V_g], \quad u = 0, 1, 2, \dots, \text{number of users} \quad (2)$$

where V_u represents a vector which holds all the user registrations, user_id denotes an identifier unique for each user and V_g is a pointer to (1) which shows the service class that could be supported.

The service class is a factor corresponding to the number of services that the requested sensor node can serve upon a grid call arrival. As mentioned in the previous section, a sensor device can support multiple services at the same time, each associated with a transceiver, e.g. light, sound and humidity monitoring, temperature and vibration sensing. Each class denotes the number of services that can be served at the same time from a requested node. Without loss of generality, we assume for the rest of the paper that sensor node measurements for all kinds of supported services require the same power consumption level. A typical example of QoS classification and average resource consuming estimation is presented in Table 2, where five of the most popular sensor activities have been taken into consideration.

Table 2. Service class characteristics

QoS Classification	Types of provided services by sensor devices	Resource Utilization (%) of sensor device per measurement
Class 0	Light Monitoring, Sound Monitoring, Humidity Monitoring, Temperature Sensing and Vibration Sensing	100
Class 1	Light Monitoring, Sound Monitoring, Humidity Monitoring, Temperature Sensing	80
Class 2	Light Monitoring, Sound Monitoring, Temperature Sensing	60
Class 3	Light Monitoring, Temperature Sensing	40
Class 4	Temperature Sensing	20

Other sensor activities, such as pollution measurements, could also be adapted to the proposed model. However, for simplicity reasons, the above mentioned five well-known sensor activities have been considered for the proxy-based middleware framework. The percentage of the average resource utilization accounts for specific energy thresholds that can be observed in a sensor node. Based on the requested

service class thresholds and the information of the remaining power resources of each node in the given sensor area, the scheduler (communicating with the resource manager) classifies the availability of services and accepts or denies the grid user requests according to the network status.

Note that in this paper we have focused on the energy efficiency for middleware sensor grid deployments, hence all scheduler operations such as user authentication and authorization, grid call acceptance or drop call events, service and user group classification have been implemented from a power awareness point of view. Hence, the proposed scheme is termed as *power efficient*.

3.4 Resource Manager

In order to serve a valid grid request, a second-level control mechanism checks the current network status and the availability of the requested sensor nodes. The main role of the resource manager is the energy conservation of the entire monitored sensor network. The routing decision from the requested node/nodes to the gateway is taken according to the remaining energy of the sensor nodes and the energy consumption for the specific service class demands. Therefore, the resource manager finds optimal [source/destination] paths while extending the sensors’ lifetime and preventing network partitioning. In particular, it keeps a record file containing all the monitored sensor ids with their respective remaining energy resources. It is also aware of all the adjacency links among the nodes within its sensor area.

For modeling purposes we consider a rectangular grid area where all the nodes are placed uniformly in the plane. Hence, the grid area is a $[n \times n]$ matrix, where n denotes the number of nodes in the sensor area. Let $AdjL = [n^2 \times n^2]$ matrix corresponding to all available links among neighboring nodes. We define the following indicator function:

$$AdjL_{ij} = \begin{cases} 1, & \text{if } i \text{ is adjacent to } j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $AdjL_{ij}$ denotes the existence of an adjacent link from node i to node j . We also define:

$$P_c^j = N_s^c \hat{p}^j \tag{4}$$

where \hat{p}^j denotes the power consumption for a single service measurement for node j , N_s^c denotes the maximum number of services of class c and P_c^j denotes the overall energy consumption for node j for service class c . Hence, each routing path can be expressed as:

$$Path_{\xi} = \sum_{j=1}^m [P_c^j], \quad \xi \in \mathbf{Z}^+ \tag{5}$$

with respect to the power consumption within sensor area, where ξ stands for the number of available paths in a sensor area and m denotes the number of hops from the gateway to node j or vice versa. In order to optimize the overall energy conservation, the resource manager always selects the shortest path/paths since it

maintains low-level energy thresholds in its monitored sensor area. Thus, it selects all the paths with minimum hop count m , as:

$$Sh_P_x[] = \sum_{X \subseteq \Omega} \min_m (Path_x) \tag{6}$$

where Sh_P_x stands for the shortest path selection with respect to X which denotes a subset of all available shortest paths within a subset of Ω , where Ω denotes all the available paths from the gateway to node j or vice versa.

Due to the assumption of uniform distribution of the sensor nodes the network topology may provide several shortest paths with equal hop count m , for a given node j to gateway. A typical example is shown in figure 4 where node A can communicate with node B via one of its adjacent nodes denoted with dashed line. The latter diversity in path selection is provided due to symmetry nature of the $[4 \times 4]$ grid topology.

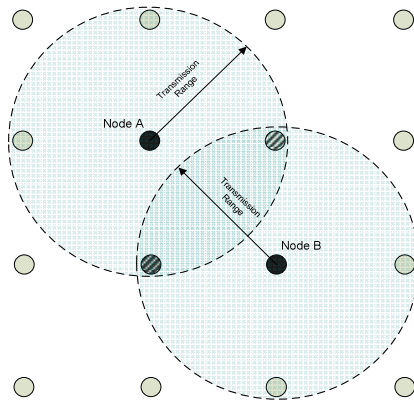


Fig. 4. Grid Topology Simple-Case Scenario

The resource manager (in collaboration with the scheduler), finds the optimal shortest path by solving the following linear program:

$$find \quad \min_{X \in X} Sh_P_x \tag{7}$$

subject to

$$[Cur_Sensor_thr]_j - [Sensor_thr]_j - P_c^j \geq 0, \quad \forall j \text{ in the path} \tag{8}$$

$$[Res_util]_u^V - [req_C]_u \geq 0 \tag{9}$$

where Cur_Sensor_thr denotes the available power resources of node j and $Sensor_thr$ denotes a minimum power level that a considered node must possess in order to participate in communication procedure. Finally, req_C denotes the requested service class upon a grid call arrival from user u .

Upon the selection of the optimal path, the resource manager sends to the gateway a sensor-id list which contains the requested sensor node and all the other nodes that participate in the selected path. In order to take into consideration the latter signaling cost, (8) is transformed to:

$$[Cur_Sensor_thr]_j - [Sensor_thr]_j - P_c^j - sig_P_c^j \geq 0, \quad \forall j \text{ in the path} \quad (10)$$

where $sig_P_c^j$ denotes the signaling cost percentage with respect to P_c^j , for a packet transmission from the gateway to node j .

If there is no available shortest path fulfilling the criteria denoted by the scheduler and the resource manager, the above mentioned linear program is re-executed substituting C_n to C_{n+1} until all the available service classes are covered, in order to minimize the drop call probability.

4 Performance Evaluation

We have implemented the proposed framework in a JDK 6.0 environment. For our simulations we consider that the grid call arrival rate from each user group is defined by a Poisson process, according to the number of requests per minute of each user group. Analytical simulation user parameters are listed in Table 3. Each call is associated with a specific node-id from the sensor grid. For the selection from the grid, sensor nodes are statistically independent, identically distributed with unit variance. We therefore used a uniform distribution for the association of requested nodes for each call due to their equal selection probability. The sensor area is considered to be a WSN consisting of 100 equally positioned nodes (10×10 grid plane) and the gateway is placed in the middle of the sensor area in order to maximize the path selection diversity and to avoid rapid network energy saturation.

Table 3. User group statistics

	Administrator	Government	Academic	Commercial	Unsubscribed
Number of Users	100	200	2000	4000	8000
Call Rate per user (per minute)	0.15	0.1	0.05	0.03	0.015
res_util (%)	100	100	80	60	20
group_thr (%)	0	5	10	15	20

We fix parameter \hat{p}^j and $sig_P_c^j$ to be $0.23 \times 10^{-3} \%$ and $0.023 \times 10^{-3} \%$ respectively, according to the methodology followed in [10-12]. For the configuration of the proposed resource manager component (as described in subsection 3.4), we have implemented a breadth-first search in order to find the optimal shortest path/paths. A crucial benefit for the above mentioned decision is its direct response mainly due to the low-level complexity of the algorithm defined as $O(|V|+|E|)$. Figure 5, shows the flowchart of the proposed scheme and its basic characteristics.

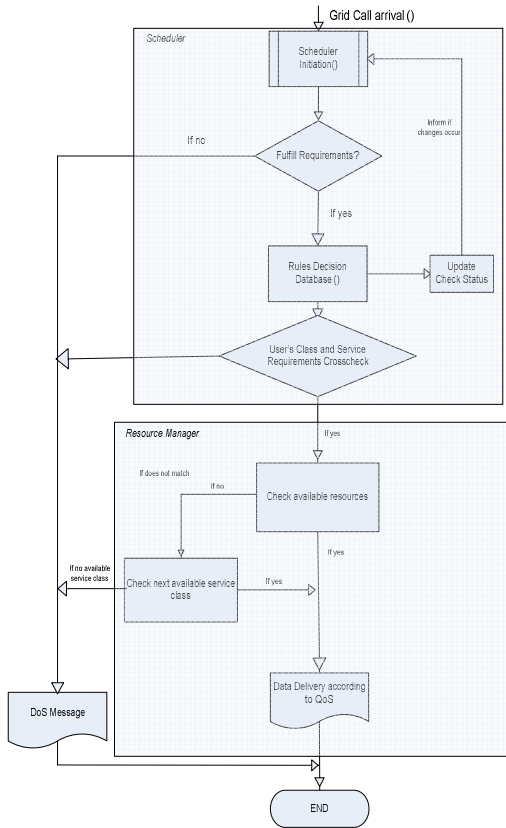


Fig. 5. Flowchart of the proxy-based Middleware

4.1 Simulation Results

In order to evaluate the effectiveness of the proposed scheme, a cross-reference scenario was necessary. We therefore compare our scheme to a secondary algorithm (it is termed *Res_Algorithm*) which employs only the resource manager component. Since scheduler is not adopted, it can not filter grid calls according to the “type of user” criterion. In other words, it does not provide any QoS classification. All user groups can get access to all available service classes. In addition, the resource manager of the secondary algorithm is not optimized, in the sense that it does not implement the linear program in (7), (9) and (10) for the shortest path selection. Instead, it selects randomly one of the shortest paths derived by (6). A tertiary algorithm (called *Sched_Algorithm*) employs the proposed scheduler component and the modified simple resource manager component as illustrated in the *Res_Algorithm*.

The main goal of the implementation of the two alternative schemes is to evaluate independently the importance of the proposed scheduler and the resource manager components.

Figure 6 shows the aggregated percentage of the energy consumption of the entire WSN as a function of the simulation time. The termination of the energy consumption

lines means that there is no more available energy to serve requests. As expected the *Res_Algorithm* results in the shortest network lifetime because the absence of the Scheduler leads to much higher energy consumption for serving the requests of the less-privileged user groups. We can also observe that the energy consumption exhibits extremely sharp falls due to the fact that the Resource Manager does not perform the linear programming of functions 7-9. The result is that the energy is consumed linearly until it is enough only for low service classes and then until its complete exhaustion. Although the *Sch_Algorithm* controls better the energy consumed by the less privileged user groups, it still suffers from the sharp falls that shorten the network lifetime. The proposed architecture leads to the longest serving duration due to the more sophisticated implementation of the Resource Manager. When the available energy falls below 30%, the Resource Manager accepts requests from increasingly fewer user groups to reserve power for the more privileged users. Consequently the energy consumption degrades more gracefully and the lifetime of the sensor network is prolonged in favor of the more “important” users.

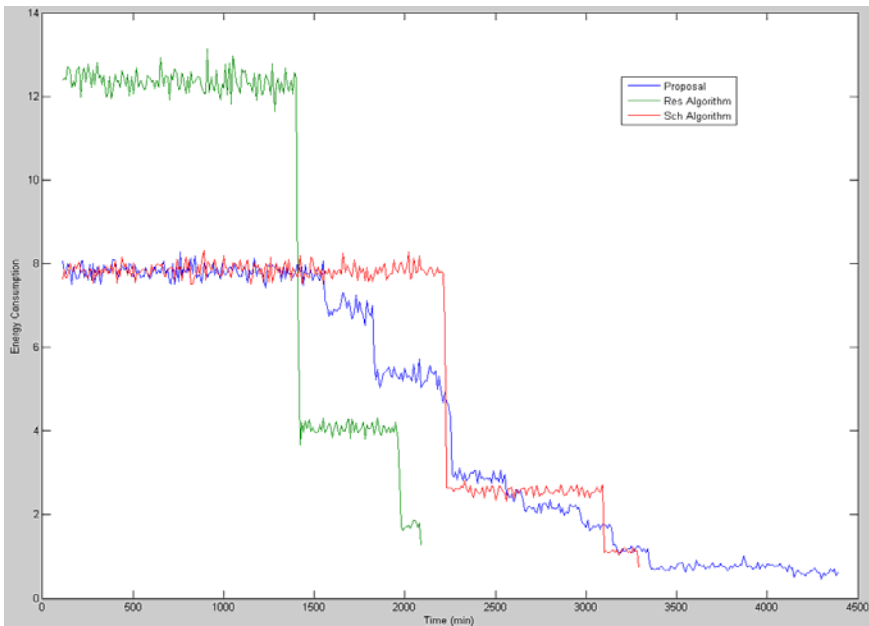


Fig. 6. The aggregated energy consumption percentage of the three simulated algorithms as function of the time

Figure 7 shows in more detail how the Proposed Algorithm affects the energy consumption and the accepted request rate per user group in comparison to the *Res_Algorithm*. When the *Res_Algorithm* is used, the percentage of the accepted requests and the energy consumed by a user group are connected through the equation

$$\frac{E_i}{\sum E_i} \approx \frac{R_a^i}{\sum R_a^i} \approx \frac{R_i^i}{\sum R_i^i},$$

where E_i is the energy consumption, R_a^i is the rate of the

accepted requests from group i and R_i^j is the overall request rate from group i . Namely the amount resources that a user group utilizes depends to its request rate and not to its role. As figure 7 indicates, the Proposed Algorithm achieves the desired QoS differentiation by accepting fewer requests from the Commercial and Unsubscribed user groups, giving priority to the more privileged users. As a result, the acceptance rates of the Administrator and the Governmental users increase by 132% and 46.5% respectively. The energy consumption change ratio is even larger because of the consumption restrictions that the Scheduler imposes to the lower user groups.

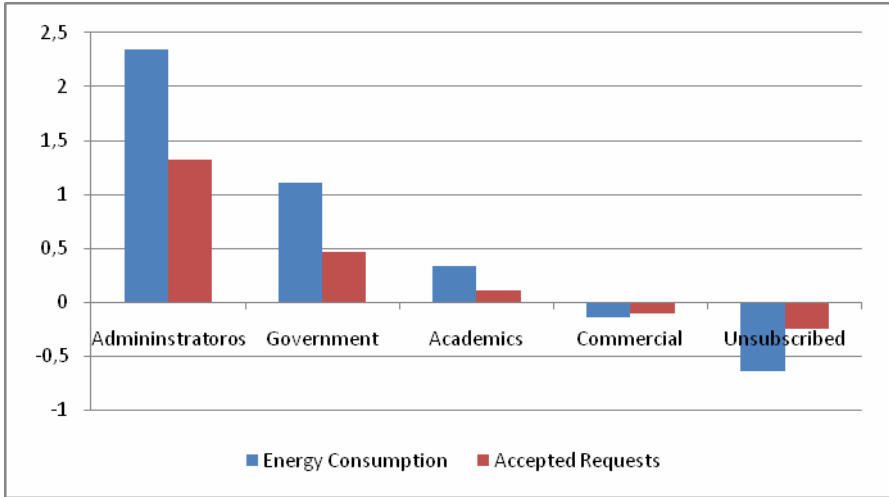


Fig. 7. The change ratio of the energy consumption percentage and the accepted request rate between the Proposed Algorithm and the Res_Algorithm

Although the simulation results illustrate the benefits from using the Resource Manager in conjunction with the Scheduler, the above scenario is only indicative. The logic of the Resource Manager and the Scheduler can be easily expanded to meet the requirements of different grid applications. As an example of such optimizations we can simulate a second test case that is similar to the previous but now we assume that the grid applications are not extremely time-sensitive. In this case the Scheduler can use a delay queue that caches the incoming requests for a short time interval before forwarding them to the Resource Manager. If there are multiple requests for the same node before the delay time timeouts, these requests can be translated to only one sensor-level request. The delay time can be a multiple of the interarrival request time and if a request is time-sensitive, a flag can be set to indicate that it should be served directly. Figure 8 depicts the performance of this delay queue with respect to the total number of accepted requests when using the Proposed Algorithm. We should note that for 19.5% consumed energy the network cannot serve more requests. The reason is that the nodes that communicate with the sink have consumed all their energy while the nodes' energy increases as we move further from the sink.

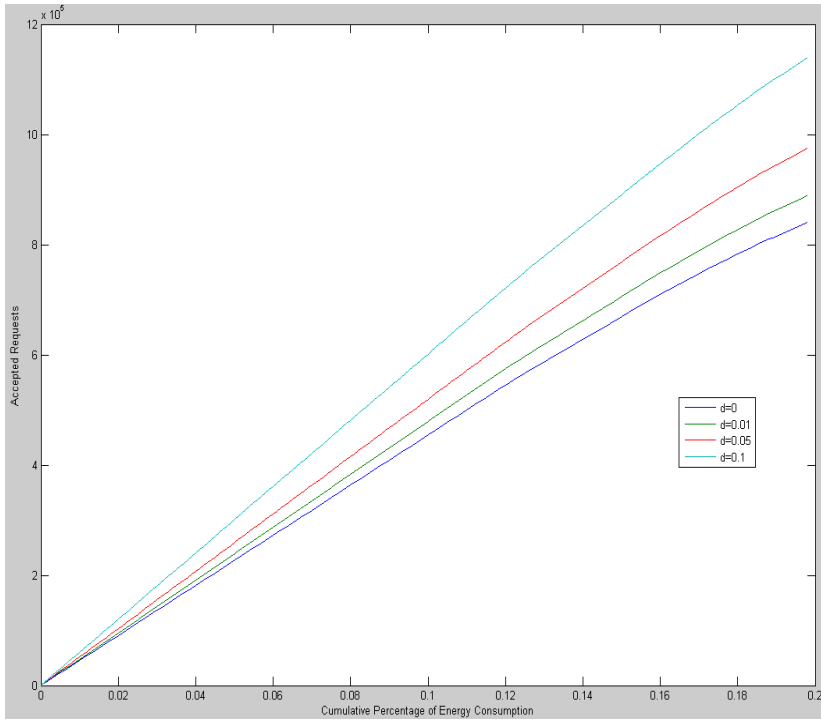


Fig. 8. The total number of accepted requests as a function of the cumulative percentage of consumed energy of the whole WSN

As expected, longer delays entail a larger number of served requests for the same level of energy consumption. However the choice of the appropriate delay depends on how timely an application should be. It should be also noted that the performance of the queue depends on two important parameters, the number of the nodes and the probability distribution for selecting a specific node. When the nodes are selected uniformly the delay queue performs worse for larger networks. On the contrary the efficiency of the delay queue increases if some nodes have a higher probability to be selected than others (e.g. following the Binomial distribution). This is usual if some phenomena take place only in specific areas of the WSN. Finally, it is worth mentioning that the delay queue allows the less privileged user groups to obtain measurements even if the available energy is below their respective energy threshold. This can happen whenever two different requests for the same node are generated by two different user groups, a privileged and a less privileged one. The Scheduler will be responsible to extract the data from the reply to the privileged user group, that are allowed to be accessed by the less privileged.

4.2 Discussion

The major advantages of our work are as follows. Firstly, it is a scheme which combines efficient routing in WSN infrastructures and QoS classification in a

personalization logic basis from an energy awareness perspective. Additionally, it is the first scheme which combines the above mentioned characteristics in an integrated platform designed specifically for sensor grid applications. Secondly, it is fully compatible with any sensor network deployment, in the sense that is placed in the middleware without any interaction with the MAC implementation of each WSN. It is also a distributed approach as each sensor area is managed separately by its associate proxy-based middleware. Thirdly, the low-level complexity of our scheme provides an essential benefit which is more than a prerequisite for a QoS-provisioned grid infrastructure, consisted of a dense grid community.

5 Conclusion

WSNs and Grid Computing are two promising technologies and both have been adopted into industry recently. Sensor grid deployments enhance a great potential of these technologies into a merged framework and due to that the research community has focused on innovative strategies to the field. A middleware architecture platform is a prerequisite for sensor grids in order to efficiently come through aggregated grid services and rapid user demands. In this paper, we discussed the most challenging issues for a proxy-based middleware scheme in order to cope with sensor grids and we also proposed a model which accepts grid calls according to specific service classes giving appropriate QoS on a personalization logic basis. Furthermore, the whole service handling and management framework is considered to be power-aware according to sensor network status.

References

1. Tham, C.-K., Buyya, R.: SensorGrid: Integrating Sensor Networks and Grid Computing. Special Issue on Grid Computing (2005), <http://www.gridbus.org/reports/sensor-grid.pdf>
2. Pantazis, N., Vergados, D.D., Miridakis, N.I., Vergados, D.J.: Power control schemes in wireless sensor networks for homecare e-health applications. In: Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments, Athens, Greece (2008)
3. Lim, H.-B., Teo, Y.-M., Mukherjee, P., Lam, V.-T., Wong, W.-F., See, S.: Sensor Grid: Integration of Wireless Sensor Networks and the Grid. In: The IEEE Conference on Local Computer Networks, pp. 91–99 (2005)
4. Franke, H.A., Koch, F.L., Rolim, C.O., Westphall, C.B., Balen, D.O.: Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing. In: Third International Conference on Wireless and Mobile Communications (2007)
5. Rao, I., Imran, N., Khan, S., Huh, E.-N., Chung, T.: Adaptive and Reconfigurable Resource Management for Wireless Sensors using Grid Technology. In: 2nd International Conference on Communication Systems Software and Middleware, COMSWARE, pp. 1–5 (2007)
6. Gaynor, M., Moulton, S.L., Welsh, M., Lacombe, E., Rowan, A., Wynne, J.: Integrating Wireless Sensor Networks with the Grid. *IEEE Internet Computing* 8(4), 32–39 (2004)

7. Stathis, K., Kafetzoglou, S., Papavassiliou, S., Bromuri, S.: Sensor Network Grids: Agent Environments Combined with QoS in Wireless Sensor Networks. In: Third International Conference on Autonomic and Autonomous Systems (2007)
8. YuJie, Y., Shu, W.: The key research on integrating wireless sensor network with grid. In: Proceedings, International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 1477–1480 (2005)
9. Miridakis, N.I., Vergados, D.D., Anagnostopoulos, I., Douligeris, C.: A discussion on proxy-based middleware for Sensor Grid Architectures. In: 4th EGEE User Forum/OGF25 & OGF-Europe's 2nd International Event, Catania, Sicily, Italy (2009)
10. Crossbow, MICA2 wireless measurement system. Datasheet available from, http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf
11. Piotrowski, K., Langendoerfer, P., Petter, S.: How public key cryptography influences wireless sensor node lifetime. In: The proceedings of 4th ACM Workshop on Security of ad hoc and Sensor Networks, pp. 169–176 (2006)
12. Bouabdallah, F., Bouabdallah, N., Boutaba, R.: On Balancing Energy Consumption in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology* 58(6) (2009)
13. Gaynor, M., Moulton, S.L., Welsh, M., LaCombe, E., Rowan, A., Wynne, J.: Integrating Wireless Sensor Networks with the Grid. *IEEE Internet Computing* 8(4), 32–39 (2004)
14. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., Vanderbilt, P.: Open Grid Services Infrastructure (OGSI) Version 1.0, Global Grid Forum (2003)

Supporting VoIP Services in IEEE 802.11e WLANs*

Jeonggyun Yu¹, Munhwan Choi², Daji Qiao³, and Sunghyun Choi²

¹ Samsung Electronics Co., LTD, Suwon, Korea

² The School of Electrical Engineering and INMC,
Seoul National University, Seoul, 151-744, Korea

³ Department of Electrical and Computer Engineering,
Iowa State University, Ames, IA 50011, USA

jeonggyun.yu@samsung.com, mhchoi@mwsl.snu.ac.kr, daji@iastate.edu,
schoi@snu.ac.kr

Abstract. Voice over Internet Protocol (VoIP) over Wireless Local Area Network (WLAN) is becoming popular thanks to its cost efficiency. However, it has been a challenge to provide good quality of VoIP services in WLANs, which is due mainly to (i) the nature of contention-based channel access of WLAN Medium Access Control (MAC); (ii) the presence of coexisting non-real-time data traffic; and (iii) the time-varying WLAN capacity caused by transmission rate diversity and variation of stations over time. In this paper, we propose a simple, effective and viable solution to improve the quality of VoIP services in 802.11e contention-based WLANs, which basically utilizes the advanced features of 802.11e MAC for QoS support. The key ingredients of our solution include (i) a priority queue to serve the VoIP traffic with higher priority than the non-real-time data traffic; and (ii) a conservative history-based admission control scheme for VoIP services, which accommodates the transmission rate diversity and variation of ongoing VoIP sessions over time. Simulation results demonstrate that our solution admits as many VoIP calls as possible without compromising the quality of their services.

Keywords: IEEE 802.11e EDCA, VoIP, QoS.

1 Introduction

Voice over IP (VoIP) and IEEE 802.11 Local Area Network (WLAN) have seen tremendous growth in recent years. IEEE 802.11 WLAN [1] has become the dominant technology for indoor broadband wireless networking. VoIP has been widely adopted in the enterprise and residence environments thanks to the various advantages such as a lower-cost, easy setup, and the integration of voice and data networks. The emergence of many VoIP vendors and VoIP service providers such as Skype [2] also speeds up the usage of VoIP.

* This research was supported by the MKE, Korea, under the ITRC support program supervised by the NIPA (NIPA-2009-C1090-0902-0006).

How to provide high Quality of Service (QoS) for VoIP applications in 802.11 WLANs has received considerable research attention. Originally, the 802.11 WLAN was designed to support best-effort services which do not have stringent QoS requirements, such as Internet-based Non-Real-Time (NRT) data services like Web browsing, e-mail, and file transfer. Therefore, many efforts to support the QoS in legacy 802.11 WLAN have been made [4,5,6]. However, those still have inherent inefficiencies such as the lacks of admission control, QoS signaling, differentiated channel access, and so on.

The 802.11e [2], which is an amendment to the legacy 802.11 Medium Access Control (MAC), was designed with the aim to support QoS [7,8,9]. The 802.11e MAC is expanding the 802.11 application domain by enabling Real-Time (RT) services such as voice and video services. The 802.11e MAC protocol is called the Hybrid Coordination Function (HCF), which contains a contention-based channel access mechanism (EDCA). EDCA is an enhanced version of the legacy Distributed Coordination Function (DCF) for QoS support. Most of the off-the-shelf 802.11e-compliant products, which are certificated by the Wi-Fi alliance [10], implement EDCA.

However, even when the 802.11e EDCA is employed, there are still some challenges to provide high-quality VoIP service in WLANs as follows: (i) the difficulty in quantitatively controlling channel occupancy of stations due to the contention-based channel access of EDCA; (ii) the presence of coexisting NRT data traffic; and (iii) the time-varying WLAN capacity caused by transmission rate diversity and variation of stations over time.

In order to address the above issues, we propose an effective and standard-compliant solution for improving the quality of VoIP services in 802.11e contention-based WLANs. It utilizes the advanced features in the 802.11e such as service differentiation mechanism and admission control framework, and consists of the following components: *Priority Queuing (PQ)* and *Call Admission Control (CAC)*.

(1) *Priority Queuing (PQ)*: Different from the legacy 802.11 MAC with a single First-In-First-Out (FIFO) transmission queue, an EDCA MAC contains multiple queues with different channel access priorities. This means that, when a WLAN carries a mixed traffic of voice and NRT data packets, the 802.11e EDCA MAC is able to provide differentiated services to VoIP applications which have stringent QoS requirements. The idea of PQ is to give voice queue strictly higher priority than NRT data queue by assigning proper channel access parameters to each of the queues so that NRT data queue can access the channel only when RT queue is empty. Moreover, in order to mitigate the bottleneck issue at the AP, which limits the VoIP capacity [11], we adopt a simple contention-free access scheme for the AP (called PIFS Access) by controlling channel access parameters of its AC_VO queue.

(2) *Call Admission Control (CAC)*: One of the key elements in improving the quality of VoIP services is effective call admission control, which determines whether to admit a new VoIP call based on the available capacity of the WLAN,

so as to maintain the QoS of admitted VoIP calls while accommodating as many new calls as possible [12]. However, it is nontrivial to obtain an accurate estimation of the available WLAN capacity. This is because the link conditions between the AP and stations fluctuate due to multipath fading and/or user mobility. In this paper, we propose an admission control scheme based on the framework provided in the 802.11e standard. The proposed scheme predicts the future transmission rates of ongoing VoIP sessions based on their transmission histories and then calculates the expected amount of VoIP service time to determine the admission of a new VoIP call. Moreover, it limits the channel occupancy times of the admitted VoIP sessions by assigning the maximum allowable channel access time to each admitted VoIP session, which is referred to as *Medium Time (MT)* and derived based on its QoS requirements and transmission rate.

The rest of this paper is organized as follows. Section 2 introduces the EDCA admission control framework and discusses the necessities of admission control for VoIP services over IEEE 802.11e WLAN. The details of the proposed solution are described in Section 3. Section 4 presents the simulation results, and the paper concludes in Section 5.

2 TSPEC and EDCA Admission Control for VoIP Services

In an IEEE 802.11e WLAN, a VoIP station sets up a virtual connection, called *Traffic Stream (TS)*, with the AP before commencing any actual voice packet transfer in order to provide the prescribed QoS for its VoIP call. The admission controller located at the AP determines whether to admit a new VoIP call based on the available capacity of the WLAN and the QoS requirement of the VoIP call. If the new VoIP call is admitted, the corresponding VoIP TS is set up between the new VoIP station and the AP.

The QoS requirement and traffic characteristics of a TS, called *Traffic Specification (TSPEC)*, usually can be provided from the application layer, e.g., VoIP application, via station management entity (SME), which is a cross-layer entity and can internally communicate with multiple protocol layers. The TSPEC is submitted to the admission controller located at the AP by a station when it requests the admission of its VoIP call and wants to set up the corresponding TS. Then, the TSPEC is used by the admission controller to make the admission decision.

Fig. 1 shows the admission control and VoIP TS setup procedure for VoIP services, which is based on the 802.11e standard [2]. An ADDTS Request frame, which conveys the TSPEC element, is transmitted by a VoIP station to the AP in order to request a VoIP TS setup. The TSPEC consists of several parameters like Nominal MSDU Size, Mean Data Rate, Delay Bound, Minimum PHY Rate, Medium Time, and so on. Most of parameters except *Medium Time (MT)*, which is the amount of time allowed for the corresponding VoIP station to access the medium per one-second period, are specified by a VoIP station and delivered to the AP when it requests the admission of its VoIP call. If the AP decides to

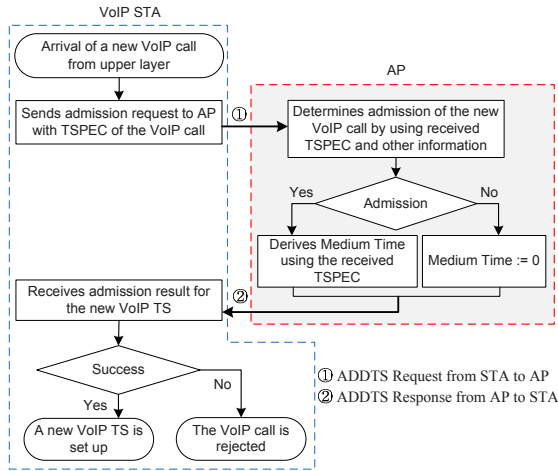


Fig. 1. Traffic Stream (TS) setup and EDCA admission control procedure for VoIP services

accept the request, the AP derives the MT from the parameters conveyed in the TSPEC element of the ADDTS request frame. Then, the AP sends the derived MT to the requesting VoIP station via an ADDTS Response frame.

After receiving the ADDTS Response frame, the admitted VoIP station i records the MT as Admitted Time \mathcal{A}_i by $\mathcal{A}_i = \mathcal{M}_i \cdot t_a$, where \mathcal{M}_i is the MT of VoIP TS i and t_a is an averaging period. \mathcal{A}_i represents the maximum amount of time that the station can use to transmit packets belonging to the corresponding VoIP TS within every t_a -second time window, where a design parameter t_a ($1 \leq t_a \leq 100$) is an integer [2]. For each packet transmission, a VoIP station increases the *Used Time* (\mathcal{U}), which is the amount of time used to attempt VoIP packet transmissions, and if \mathcal{U}_i is larger than or equal to \mathcal{A}_i , VoIP station i cannot transmit more voice packets via AC_VO until the next t_a interval [1].

Now, the problems left for admission control are as follows: (i) how to decide the admission of a new VoIP call (i.e., a VoIP TS); and (ii) how to derive the MT of admitted VoIP TS. The proposed solutions to these problems are presented in Section 3.

3 Proposed Solution for Improving Quality of VoIP Services in 802.11e EDCA

Our objective is to improve the quality of VoIP services in IEEE 802.11e WLANs. To achieve this goal, we propose a solution that implements the following modules at the AP.

¹ Actually, the VoIP station may transmit voice packets via other Access Categories (ACs) where no admission control is required such as Best Effort Access Category (AC_BE). However, in this paper, we assume that voice packets are transmitted via the Voice Access Category (AC_VO) only.

- *Priority Queueing via Controlling Channel Access Parameters* to provide service differentiation between voice traffic (i.e., AC_VO) and NRT data traffic (i.e., AC_BE); and
- *Call Admission Control* to control the admission of new VoIP calls into the network and to efficiently allocate MTs to admitted VoIP sessions.

Fig. 2 shows the system architecture of the AP with our proposed solution. A packet from the upper layer can be classified into one of four ACs based on various parameters such as Ethernet, TCP/IP, and IEEE 802.1D/Q parameters [2, 13].

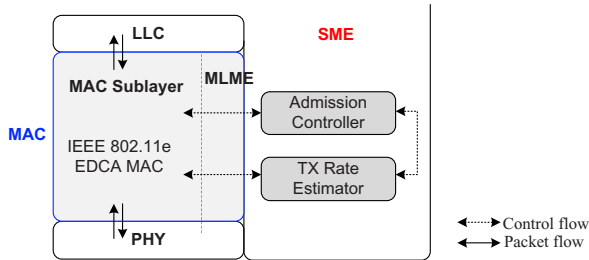


Fig. 2. Overview of the proposed solution. MLME is MAC Layer Management Entity.

TX Rate Estimator caches the time used to attempt the transmission of a voice packet belonging to each admitted VoIP session, including the wasted time for a failed transmission attempt. When a new VoIP call requests its admission, *TX Rate Estimator* estimates the future transmission rates of all the currently ongoing VoIP TSs based on the cached history. The estimation algorithm is presented in Section 3.3. Based on the rate estimation and an assumption that the new VoIP call transmits at its minimum PHY rate if admitted, *Admission Controller* calculates the total VoIP service time, determines the admission of the new VoIP call, and then calculates the MT of each admitted VoIP TS. Moreover, while calculating the total VoIP service time, it determines the channel access parameter values for AC_VO of stations that can minimize the channel time occupied by VoIP traffic, thus increasing the channel utilization for AC_BE (i.e., NRT traffic). Then, the AP updates channel access parameters of stations and MTs of all the admitted VoIP TSs. The calculation details of the VoIP service time and admission control algorithm is presented in Section 3.3.

3.1 System Model and Assumptions

We consider G.711 [14] – the simplest voice codec. Note that our analysis and design could be applied to other voice codecs as well. The G.711 codec generates a 64 kbps data stream, based on an 8-bit Pulse Coded Modulation (PCM), with the sampling rate of 8000 samples/s. We assume that Voice Activity Detection (VAD) is not used, which means that the VoIP traffic is a CBR (Constant Bit

Table 1. IEEE 802.11b PHY and VoIP Parameters

Parameter	Values
Slot Time (σ)	20 μ s
SIFS	10 μ s
PIFS	30 μ s
PHY Overhead (\mathcal{O}_{PHY})	192 μ s
MAC Overhead (\mathcal{O}_{MAC})	30 bytes
ACK Length (\mathcal{L}_{ACK})	14 bytes
Voice Packet MSDU Size ($\mathcal{L}_{\text{voice}}$)	Voice Data (160) + RTP Header (12) + UDP/IP Headers (28) + SNAP Header (8) = 208 bytes
Mean Data Rate (ρ_{voice})	208 bytes / 20 ms = 83.2 kbps

Rate) traffic. We assume that a voice packet is generated every 20 ms. Thus, the amount of voice data carried in a packet is 160 bytes = 8000 samples/s \times 8 bits/sample \times 20 ms. Real-Time Protocol (RTP) over User Datagram Protocol (UDP) is usually used for the VoIP transfer. When an IP datagram is transferred over the 802.11 WLAN, it is typically encapsulated by the IEEE 802.2 Sub-Network Access Protocol (SNAP). Accordingly, the size of a voice packet at the 802.11 MAC Service Access Point (SAP)² i.e., the voice packet MSDU (MAC Service Data Unit) size, is 208 bytes, as shown in Table 1.

3.2 Priority Queuing via EDCA Parameter Setting

In this paper, we consider only AC_VO and AC_BE out of four ACs for simplicity. For service differentiation between AC_VO and AC_BE, we use the strict priority queuing by properly setting the channel access parameters of the two ACs. Moreover, we adopt a simple contention-free access (called PIFS access) for the AP's AC_VO [15], which allows the AC_VO of the AP to transmit a pending voice packet after a PIFS idle time without any contention.

Parameter Setting for AC_VO. In the 802.11e standard [2], AIFSN[AC_VO], which is the arbitration interframe space number for AC_VO, is an integer greater than 1 for stations and an integer greater than 0 for the AP [2]. Moreover, the values of minimum and maximum contention window size for AC_VO, i.e., $CW_{\min}[\text{AC_VO}]$ and $CW_{\max}[\text{AC_VO}]$, can be set to zero. Therefore, for AC_VO of the AP, we can use the smallest access parameter values of $\text{AIFSN}[\text{AC_VO}] = 1$ and $CW_{\min}[\text{AC_VO}] = CW_{\max}[\text{AC_VO}] = 0$ for downlink voice packet transmissions. This allows the AC_VO of the AP to transmit the pending voice packets after a PIFS idle time without backoff. This scheme is referred to as *PIFS Access* for the rest of this paper.

² MAC SAP is the interface between the MAC and the higher layer, i.e., the IEEE 802.2 Logical Link Control (LLC) layer.

On the other hand, AC_VO of a station uses $AIFSN[AC_VO] = 2$, which is the smallest value for a station, and both $CW_{\min}[AC_VO]$ and $CW_{\max}[AC_VO]$ are set to a properly chosen value based on the given parameters, i.e., the number of VoIP stations and their transmission rate distribution. How to find the proper $CW_{\min}[AC_VO]$ value is presented in Section 3.3. $CW_{\max}[AC_VO]$ uses the same value as $CW_{\min}[AC_VO]$ so that delay and delay jitter performance of VoIP traffic can be improved without doubling the contention window size after a transmission failure.

Parameter Setting for AC_BE. In order to prevent AC_BE from accessing the channel while AC_VO has any voice packet to transmit, an AC_BE uses $AIFS[AC_BE]$ value as follows:

$$AIFS[AC_BE] = AIFS[AC_VO] + CW_{\min}[AC_VO] \cdot \sigma, \quad (1)$$

where σ is a backoff slot time and $AIFS[AC_BE]$ and $AIFS[AC_VO]$ are the arbitration interframe space for AC_BE and AC_VO, respectively. Therefore, after a channel busy period, an AC_BE can start its backoff only if there is no AC_VO with pending packets. For $CW_{\min}[AC_BE]$ and $CW_{\max}[AC_BE]$, the default values provided in the standard [2] are used.

3.3 Conservative Admission Control for VoIP Services (CAVS)

We propose a history-based admission control scheme, called CAVS (Conservative Admission control for VoIP Services), to accommodate the transmission rate diversity and variation of VoIP stations over time. The key ideas of CAVS are (i) caching the recent transmission results of the admitted VoIP sessions; (ii) determining the admission of a new VoIP call based on a conservative history-based estimation of future transmission rates of the admitted VoIP sessions; and (iii) deriving/updating MTs of the admitted VoIP stations.

Transmission History Cache. In CAVS, whenever the AP finishes a packet transmission attempt, it caches the result as a quadruplet

$$\Theta : (t_{\text{event}}, \text{session_id}, \text{result}, \text{time_usage}), \quad (2)$$

where t_{event} is the time instance when the AP started to transmit the packet, session_id is the ID of the corresponding VoIP session, result is 1 if the transmission was successful, 0 otherwise, and time_usage is the time used to complete the transmission attempt, i.e.,

$$\text{time_usage} = \begin{cases} T_{\text{voice}} + \text{SIFS} + T_{\text{ack}}, & \text{if TX success,} \\ T_{\text{voice}} + \text{ACKTimeout}, & \text{if TX failure,} \end{cases} \quad (3)$$

where T_{voice} and T_{ack} are the transmission durations of a voice packet and a ACK frame, respectively. The AP caches the recent transmission results of the admitted VoIP TSSs, and removes stale data from its cache. More specifically, the AP only caches results with $t_{\text{event}} > t_0 - t_{\text{win}}$, where t_0 is the current time and a design parameter t_{win} is the estimation window size. Based on this cached information, when a new VoIP call requests its admission, the AP estimates the future transmission rates of the admitted VoIP TSSs.

Table 2. Time used to complete a successful transmission attempt of a voice packet at each rate of the 802.11b PHY

TX Rate r^* (Mbps)	1	2	5.5	11
$T_{\text{succ}}(r^*)$ (μs)	2394	1394	793	622

Transmission Rate Estimation. In CAVS, when a new VoIP call requests its admission, the AP first calculates the average time used to attempt a successful voice packet transmission for each admitted VoIP session, based on the cached information. For the admitted VoIP session i , it is $T_{\text{avg},i} = \frac{T_{\text{total},i}}{N_{\text{succ},i}}$, where

$$\begin{cases} T_{\text{total},i} = \sum_{\substack{t_0 - t_{\text{win}} < \Theta.t_{\text{event}} \leq t_0 \\ \Theta.\text{session_id} = i \\ \Theta.\text{result} = 1}} \Theta.\text{time_usage}, \\ N_{\text{succ},i} = \sum_{\substack{t_0 - t_{\text{win}} < \Theta.t_{\text{event}} \leq t_0 \\ \Theta.\text{session_id} = i}} \Theta.\text{result}. \end{cases} \tag{4}$$

Then using $T_{\text{avg},i}$, the AP estimates the future transmission rate of the admitted VoIP session i as follows:

$$r_{\text{next},i} = \begin{cases} \min \{r_m^*, r_{\text{curr},i}\}, & \text{if } T_{\text{avg},i} \leq T_{\text{succ}}(r_{m-1}^*), \\ \min \{r_j^*, r_{\text{curr},i}\}, & \text{if } T_{\text{succ}}(r_j^*) < T_{\text{avg},i} \leq T_{\text{succ}}(r_{j-1}^*), \\ & \text{where } j = 2, \dots, m-1, \\ \min \{r_1^*, r_{\text{curr},i}\}, & \text{if } T_{\text{avg},i} > T_{\text{succ}}(r_1^*), \end{cases} \tag{5}$$

where m is the number of available transmission rates, $r_{\text{curr},i}$ is the current transmission rate of VoIP session i , and $T_{\text{succ}}(r^*)$ is the time used to complete a successful transmission attempt of a voice packet at rate r^* :

$$T_{\text{succ}}(r^*) = T_{\text{voice}}(r^*) + \text{SIFS} + T_{\text{ack}}. \tag{6}$$

For example, for the 802.11b PHY, $m = 4$ and r_1^*, \dots, r_4^* are 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps, respectively, and the corresponding T_{succ} values are listed in Table 2.

In general, Eq. (5) works fine in predicting the future transmission rate with random station movement. However, under certain circumstances, it may not perform well. For example, if the VoIP station of the admitted session i keeps moving away from the AP, $r_{\text{est},i}$ is then not a good estimation of session i 's future transmission rate. Therefore, we consider the current transmission rate of each admitted VoIP session in the final estimation of its future rate as follows:

$$r_{\text{next},i} = \min \{r_{\text{est},i}, r_{\text{curr},i}\}, \tag{7}$$

where $r_{\text{curr},i}$ is the current transmission rate of VoIP session i .

Worst-Case Analysis of VoIP Service Time for Admission Control. Our admission control scheme requires the quantified calculation of the VoIP

service time. In this subsection, we analyze the VoIP service time during \mathcal{T}_v , which is the voice packet generation interval, under the worst-case scenario for uplink VoIP packet transmissions when the AP uses PIFS Access for its downlink VoIP packet transmissions.

We make the following assumptions to simplify the problem. First, we assume that all the uplink and downlink voice packets arrive synchronously at the WLAN. Therefore, at the time instance when voice packets arrive at the WLAN, the network is temporarily congested, which is the worst-case scenario in terms of channel contention. Second, we assume the p -persistent model for the EDCA [16], instead of using the binary exponential backoff. Third, we assume that different stations transmit at different rates and the finishing order of VoIP stations' uplink packet transmissions is in the descending order of the packet transmission rate. This leads to longer VoIP service time because the wasted time in collision is determined by the longest transmission duration among packets that are involved in the collision. Fourth, we assume that there is no hidden station in the network. Finally, to simplify the analysis, we assume that the channel condition between the AP and a station is symmetric and hence the AP uses the same transmission rate as the station for the VoIP session between them.

When the number of active VoIP sessions is equal to N , the average VoIP service time $\mathcal{S}_{\text{voice}}(N)$ can be expressed as

$$\begin{aligned} \mathcal{S}_{\text{voice}}(N) &= T_N^{\text{PIFS}} + \sum_{i=1}^N T_s(i) \\ &+ \sum_{k=1}^N ((E[N_k^{\text{col}}] + 1) E[I_k] \sigma + E[N_k^{\text{col}}] E[T_{c,k}]), \end{aligned} \quad (8)$$

where T_N^{PIFS} is the total transmission time to transmit N downlink voice packets via PIFS access, which is given by

$$T_N^{\text{PIFS}} = \sum_{i=1}^N (\text{PIFS} + T_{\text{voice}}(i) + \text{SIFS} + T_{\text{ack}}), \quad (9)$$

and $T_s(i)$ is the successful transmission time of an uplink VoIP packet belonging to VoIP session i , and it is given by

$$T_s(i) = T_{\text{voice}}(i) + \text{SIFS} + T_{\text{ack}} + \text{AIFS}[\text{AC-VO}]. \quad (10)$$

$E[I_k]$ is the average number of idle backoff slots preceding a collision or the successful transmission, $E[N_k^{\text{col}}]$ is the average number of collisions preceding the successful transmission, and $E[T_{c,k}]$ is the average collision time when the number of contending stations is k . $E[I_k]$, $E[N_k^{\text{col}}]$, and $E[T_{c,k}]$ can be derived as follows.

$E[I_k]$ is given by

$$E[I_k] = \sum_{i=0}^{\infty} iP(I_k = i) = \frac{(1-p)^{N-k+1}}{1 - (1-p)^{N-k+1}}, \quad (11)$$

where $P(I_k = i)$ is the probability that there are i idle backoff slots preceding a busy period (i.e., a collision or a successful transmission), and it is given by

$$P(I_k = i) = \left((1-p)^{N-k+1} \right)^i \left(1 - (1-p)^{N-k+1} \right), \tag{12}$$

where p is the channel access probability of a station. In this paper, we use a fixed p value of $\frac{2}{CW_{\min}[\text{AC_VO}] + 1}$. This makes a station more aggressive because the contention window size is not doubled even after a transmission failure. Accordingly, this results in a conservative estimation of the VoIP service time.

Moreover, $E[N_k^{\text{col}}]$ can be calculated by

$$E[N_k^{\text{col}}] = \sum_{i=1}^{\infty} iP(N_k^{\text{col}} = i) = \frac{1 - (1-p)^{N-k+1}}{(N-k+1)p(1-p)^{N-k}} - 1, \tag{13}$$

where $P(N_k^{\text{col}} = i)$ is the probability that there are i collisions preceding the successful transmission, and it is given by

$$P(N_k^{\text{col}} = i) = \left(\frac{1 - (1-p)^{N-k+1} - (N-k+1)p(1-p)^{N-k}}{1 - (1-p)^{N-k+1}} \right)^i \times \frac{(N-k+1)p(1-p)^{N-k}}{1 - (1-p)^{N-k+1}}. \tag{14}$$

Finally, the average collision time $E[T_{c,k}]$ is:

$$E[T_{c,k}] = \text{AIFS}[\text{AC_VO}] + \sum_{i=1}^{N-k} \sum_{j=i+1}^{N-k+1} \max(T_{\text{voice}}(i), T_{\text{voice}}(j)) P_{c,k}(i, j), \tag{15}$$

where $P_{c,k}(i, j)$ is the probability that the packets of stations i and j collide with each other, given that a collision occurs. It can be calculated by

$$P_{c,k}(i, j) = 1 / \binom{N-k+1}{2}, \quad i = 1, \dots, N-k, \quad j = i+1, \dots, N-k+1. \tag{16}$$

In Eq. (16), we assume that a collision is only caused by two stations' simultaneous transmissions because the probability that three or more stations transmit at the same time is very low [17], thus being negligible.

From now on, we analyze the parameters required to derive the medium time (MT) in Section 3.3 based on the above analysis. From Eq. (8), the average idle backoff time can be calculated by

$$\psi_N = \sum_{k=1}^N (E[N_k^{\text{col}}] + 1) E[I_k] \sigma. \tag{17}$$

The average time wasted by VoIP station i due to its collisions until it successfully transmits its voice packet can be estimated as follows:

$$\mathcal{O}_{\text{sur},i} = \sum_{k=1}^{l_i} \sum_{j=0}^{C_k} j (P_{c,k}(i))^j T_{\text{fail}}(i), \quad (18)$$

where $P_{c,k}(i)$, the probability that VoIP station i 's packet collides with another packet, given that a collision occurs when the number of contending stations is k , is given by

$$P_{c,k}(i) = \sum_{j=1, j \neq i}^{N-k+1} P_{c,k}(i, j), \quad (19)$$

and $T_{\text{fail}}(i) = T_{\text{voice}}(i) + \text{AIFS}[\text{AC_VO}]$, which is the wasted transmission time by VoIP station i when its transmission collides. $C_k = \lceil E[N_k^{\text{col}}] \rceil$ is the ceiled average number of collisions when the number of contending stations is k . VoIP station i finishes its voice packet transmission when the number of contending stations is l_i , where l_i is determined by station i 's transmission rate and the worst case scenario, i.e., a higher-rate station finishes its transmission earlier. If the number of VoIP stations with the same transmission rates is more than one, each of them has a value averaged over their \mathcal{O}_{sur} 's.

The total transmission time overlapped by the colliding VoIP stations assuming that the collision is only caused by two stations can be estimated as follows:

$$\delta_N = \sum_{k=1}^N E[N_k^{\text{col}}] E[T_{c,k}^{\text{overlap}}], \quad (20)$$

where $E[T_{c,k}^{\text{overlap}}]$, the average overlapped transmission time when a collision occurs, is given by:

$$E[T_{c,k}^{\text{overlap}}] = \sum_{i=1}^{N-k} \sum_{j=i+1}^{N-k+1} \min(T_{\text{fail}}(i), T_{\text{fail}}(j)) P_{c,k}(i, j). \quad (21)$$

Admission Decision and Optimal $\text{CW}_{\min}[\text{AC_VO}]$. After estimating the future transmission rate of each admitted VoIP session using Eq. (5), the AP calculates the average service time $\mathcal{S}_{\text{voice}}(N+1, \text{CW}_{\min}[\text{AC_VO}])$ using Eq. (8), where $(N+1)$ corresponds to N admitted VoIP TSS and one new VoIP call that requests the admission. Note that $\mathcal{S}_{\text{voice}}$ is the function of the number of admitted VoIP stations, their rate distribution, and $\text{CW}_{\min}[\text{AC_VO}]$. Here, when we calculate $\mathcal{S}_{\text{voice}}(N+1, \text{CW}_{\min}[\text{AC_VO}])$, we assume that the new VoIP TS will transmit at its minimum PHY rate if it is admitted. Before the AP decides the admission, it finds the optimal value of $\text{CW}_{\min}[\text{AC_VO}]$ (cw^*) as follows:

$$cw^* = \arg \min_{\text{CW}_{\min}[\text{AC_VO}]} \mathcal{S}_{\text{voice}}(N+1, \text{CW}_{\min}[\text{AC_VO}]), \quad (22)$$

where $\text{CW}_{\min}[\text{AC_VO}] \in [0, 1023]$ is an integer. When an ongoing VoIP session ends, the AP also needs to find cw^* that minimizes $\mathcal{S}_{\text{voice}}$ for $(N-1)$ existing

VoIP sessions in order to increase the channel utilization of AC_BEAs. cw^* is distributed to the stations via the upcoming beacon transmissions. Finally, the AP determines the admission of the new VoIP call as follows:

$$\begin{cases} \text{Admit, if } \mathcal{S}_{\text{voice}}(N+1, cw^*) < \phi_{\text{voice}} \mathcal{T}_v, \\ \text{Reject, otherwise,} \end{cases} \quad (23)$$

where ϕ_{voice} ($0 \leq \phi_{\text{voice}} \leq 1$) is the fraction of \mathcal{T}_v reserved for VoIP traffic, which is a design parameter.

Calculation and Update of MT. If the AP decides to admit a new VoIP call, it derives the MTs of the ongoing VoIP TSs as well as the MT of the new VoIP TS as follows.

$$\mathcal{M}_i^{\text{new}} = \left\lceil \frac{\rho_{\text{voice}}}{8\mathcal{L}_{\text{voice}}} \right\rceil \cdot \text{SurplusMPDUTime}_i, \quad (24)$$

where SurplusMPDUTime_i , the amount of time needed to transport a voice packet belonging to VoIP TS i including the overhead due to transmission failures, is given by

$$\text{SurplusMPDUTime}_i = \mathcal{O}_{\text{sur},i} + T_{\text{voice}}(i) + \text{SIFS} + T_{\text{ack}}, \quad (25)$$

where $\mathcal{O}_{\text{sur},i}$, calculated by Eq. (18), is the expected amount of inevitable collision time wasted by VoIP station i until it successfully transmits its voice packet.

Remind that limiting the uplink channel access time of an admitted VoIP station via its MT is for the reduction of the impact of the station's transmissions on the QoS of other VoIP TSs when the station tries to overuse the channel time due to its lower transmission rate than its estimated rate in Eq. (5) or many retransmissions due to the channel errors. Since the admission decision of the latest VoIP TS was based on the estimated rates of other admitted VoIP TSs, if some admitted VoIP stations happen to use lower rates than their estimated rates after the admission decision of the latest VoIP TS, the network might be saturated, and hence the QoS of the admitted VoIP TSs might be severely degraded. For this goal of MT, the following condition always needs to be satisfied:

$$T_{N+1}^{\text{AP}} + T_{N+1}^{\text{idle}} + \sum_{i=1}^{N+1} \mathcal{M}_i \leq 1 + \Delta_{N+1}, \quad (26)$$

where \mathcal{M}_i is the current MT of VoIP TS i , $T_{N+1}^{\text{AP}} = \left\lceil \frac{\rho_{\text{voice}}}{8\mathcal{L}_{\text{voice}}} \right\rceil T_{N+1}^{\text{PIFS}}$ is the expected amount of time allowed to the AP for its transmissions of downlink voice packets per one second, $T_{N+1}^{\text{idle}} = \left\lceil \frac{\rho_{\text{voice}}}{8\mathcal{L}_{\text{voice}}} \right\rceil \psi_{N+1}$ is the expected total amount of idle backoff time of VoIP stations per one second, and $\Delta_{N+1} = \left\lceil \frac{\rho_{\text{voice}}}{8\mathcal{L}_{\text{voice}}} \right\rceil \delta_{N+1}$ is the amount of the overlapped portion due to collisions among MTs of the admitted VoIP TSs per one second. Note that T_{N+1}^{AP} , T_{N+1}^{idle} , and Δ_{N+1} are derived based on the estimated rates of the admitted VoIP TSs by using Eqs. (9), (17), and (20), respectively.

The AP updates MTs of the ongoing VoIP TSs with the newly derived MTs from Eq. (24) in order to satisfy Eq. (26) by sending ADDTS Response frames to ongoing VoIP stations without receiving the corresponding ADDTS Request frame. Note that the MT of the new VoIP TS is delivered to the corresponding station via an ADDTS Response frame.

4 Performance Evaluation

In this section, we evaluate the effectiveness of the proposed solution by using the ns-2 simulator [18]. The simulated network topology is shown in Fig. 3, where multiple mobile VoIP stations are placed inside a square region with 160 meters on the diagonal. VoIP stations communicate with the remote voice gateway via the AP which sits at the center of the square region. Our proposed solution is implemented at the AP. VoIP stations transmit and receive voice packets only and each station carries a single traffic flow. VoIP traffic is modeled by a two-way CBR session with 208-byte MSDU size and 20 ms packetization interval (i.e., $\mathcal{T}_v = 20$ ms) according to the G.711 voice codec. We use the ITU E-model [22, 23] to assess the quality of mouth-to-ear (m2e) voice communication. It gives an overall rating R to the quality of a phone call where $0 \leq R \leq 100$. A VoIP session with $R \geq 80$ is called a *satisfactory VoIP session*.

The IEEE 802.11b PHY is used in our simulation. Table 1 lists the 802.11b PHY parameters. We assume an AWGN (Additive White Gaussian Noise) wireless channel and the background noise level is set to -96 dBm. Moreover, we use the log-distance path loss model with path loss exponent of 4, and the empirical BER (Bit Error Rate) vs. SNR (Signal-to-Noise Ratio) curves provided by Intersil [19]. We use the random waypoint model [20] to simulate the mobility of VoIP stations. The random waypoint model assumes that a user's movement follows a walk-and-pause pattern; the user chooses a random destination and moves towards it at a randomly-chosen speed less than or equal to the maximum speed in a flat restricted region. In our simulation, the maximum speed for VoIP stations is 2.5 m/s and the movement of a VoIP station is restricted within the square region shown in Fig. 3. Moreover, all stations use Automatic Rate Fallback (ARF) [21] – a widely-implemented rate adaptation scheme in WLAN devices, unless specified otherwise.

We simulate 40 VoIP sessions in the network. VoIP sessions start successively every 2 seconds from the beginning of the simulation. The call duration of each VoIP session is 30 seconds. The total simulation time is 100 seconds for each mobility scenario. All the simulation results are averaged over 50 mobility scenarios. The system parameter t_a is set to 5 – the default value provided by 802.11 specification [2]. t_{win} is set to 5 unless specified otherwise.

Fig. 4 shows the R values of a VoIP session, which is the first admitted session in a simulation, and the number of admitted VoIP sessions over simulation time with or without CAVS. In both cases, PIFS Access is used for AC_VO of the AP and the wireline delay is 150 ms. We observe that, when CAVS is used, the R values are always higher than 80. On the other hand, when CAVS is not used,

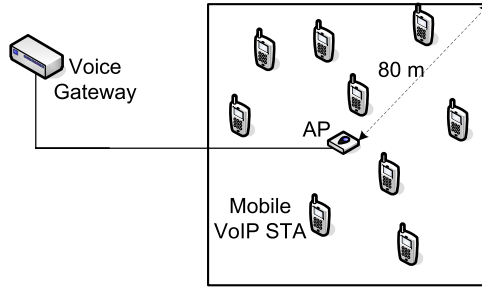
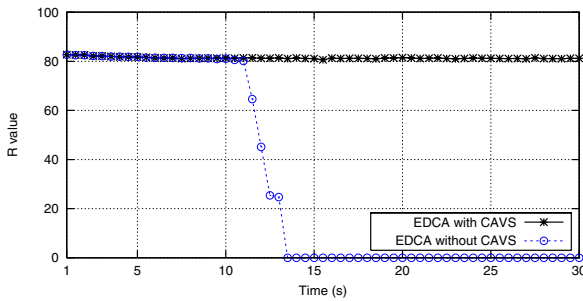
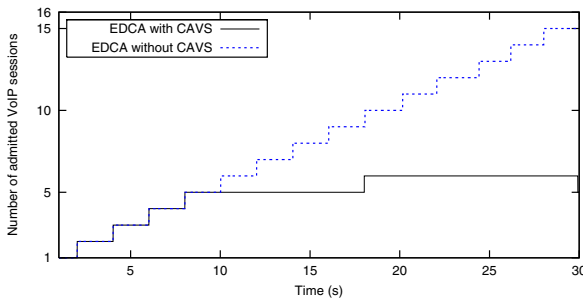


Fig. 3. The simulated network topology



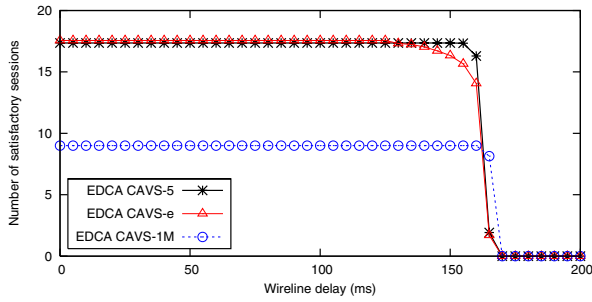
(a) R value of a VoIP session.



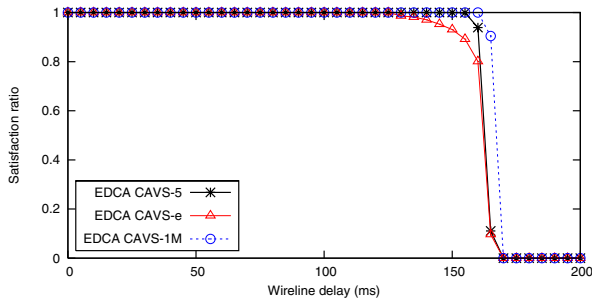
(b) The number of admitted VoIP sessions.

Fig. 4. Comparison of the cases with CAVS scheme and without CAVS scheme. The wireline delay is 150 ms.

the R values degrade severely since the network is overloaded due to the lack of admission control. As shown in Fig. 4(b), in the case of EDCA with CAVS, the number of admitted VoIP sessions is limited to 5 or 6 while, in the case of EDCA without CAVS, the number of admitted VoIP sessions increases continuously to overload the network.



(a) Total number of satisfactory VoIP sessions.



(b) Satisfactory ratio (= number of satisfactory VoIP sessions / number of admitted VoIP sessions).

Fig. 5. Comparison of CAVS schemes with different t_{win} values

To study the effect of the estimation window size (t_{win}), we simulate the following variants of CAVS:

- CAVS- t_{win} : the CAVS scheme which uses the estimation window size of t_{win} (in seconds) to estimate the future transmission rates of the admitted VoIP sessions.
- CAVS- ε : a special case of CAVS- t_{win} ; it makes an aggressive assumption that each admitted VoIP session always uses the current rate for future transmissions.
- CAVS-1M: a special case of CAVS- t_{win} ; it makes a conservative assumption that each admitted VoIP session always transmits at the lowest 1 Mbps in the future.

Fig. 5(a) shows the total number of satisfactory VoIP sessions during the 100-second simulation with different CAVS schemes. Fig. 5(b) shows the *satisfactory ratio* which is the ratio of the number of satisfactory VoIP sessions to the total number of admitted sessions. As shown in Fig. 5(b), CAVS-1M results in the perfect satisfactory ratio (i.e., 1.0) before the wireline delay gets too large to prevent any satisfactory VoIP services. However, the actual number of admitted sessions for CAVS-1M is small (i.e., 9), as shown in Fig. 5(a). This is

because CAVS-1M admits VoIP sessions very conservatively by assuming that all the admitted VoIP sessions will use the lowest transmission rate in the future. On the other hand, CAVS- ε admits VoIP sessions most aggressively among the considered schemes in our simulation. Note that, according to Eq. (7), future transmission rates of the admitted VoIP sessions estimated by CAVS- t_{win} (i.e., $r_{\text{next},i}$) are always lower than or equal to those by CAVS- ε (i.e., $r_{\text{curr},i}$). Such aggressive nature of CAVS- ε may result in incorrect admission decisions under certain circumstances, which could render a WLAN overloaded. Thus the service quality of the admitted VoIP sessions could be compromised and appears more sensitive to the increase in the wireline delay, as shown in the figures. In comparison, CAVS-5 perform better than CAVS-1M and CAVS- ε , which means that $t_{\text{win}} = 5$ is a reasonable choice for our simulated network.

5 Conclusion

In this paper, we proposed a simple, effective and viable solution to support VoIP services in 802.11e contention-based WLANs, which basically utilizes the advanced features of 802.11e MAC for QoS support. Our solution includes a priority queueing scheme to serve VoIP traffic with higher priority than non-real-time data traffic, and a conservative history-based admission control scheme for VoIP services, which accommodates the transmission rate diversity and variation of ongoing VoIP sessions over time. We evaluate our proposed solution using the ns-2 based simulation. Simulation results demonstrate that our solution admits as many VoIP calls as possible without compromising the quality of their services.

References

1. IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE std. (August 1999)
2. IEEE 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control Quality of Service Enhancements, IEEE std. (September 2005)
3. Skype. Online link, <http://www.skype.com>
4. Yu, J., Choi, S., Lee, J.: Enhancement of VoIP over IEEE 802.11 WLAN via Dual Queue Strategy. In: Proc. IEEE ICC, Paris, France (June 2004)
5. Park, E.-C., Kim, D.-Y., Choi, C.-H., So, J.: Improving Quality of Service and Assuring Fairness in WLAN Access Networks. IEEE Trans. Mobile Comput. 6(4), 337–350 (2007)
6. Zhai, H., Chen, X., Fang, Y.: A Call Admission and Rate Control Scheme for Multimedia Support over IEEE 802.11 Wireless LANs. Springer Wireless Networks 12(4), 451–463 (2006)
7. Mangold, S., Choi, S., Hiertz, G.R., Klein, O., Walke, B.: Analysis of IEEE 802.11e for QoS Support in Wireless LANs. IEEE Wireless Commun. Mag. 10(6), 40–50 (2003)
8. Choi, S., del Prado, J., Shankar, S.N., Malgoid, S.: IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation. In: Proc. IEEE ICC, Anchorage, Alaska, USA (May 2003)

9. Xiao, Y., Li, H., Choi, S.: Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs. In: Proc. IEEE INFOCOM 2004, Hong Kong (March 2004)
10. Wi-Fi Alliance. Online link, <http://www.wi-fi.org>
11. Gao, D., Cai, J., Foh, C.H., Lau, C.-T., Ngan, K.N.: Improving WLAN VoIP Capacity Through Service Differentiation. *IEEE Trans. Veh. Technol.* 57(1), 465–473 (2008)
12. Perros, H.G., Elsayed, K.M.: Call admission control schemes: A review. *IEEE Commun. Mag.*, 82–91 (November 1996)
13. Park, S., Kim, K., Kim, D.C., Choi, S., Hong, S.: Collaborative QoS Architecture between DiffServ and 802.11e Wireless LAN. In: Proc. IEEE VTC 2003 Spring, Jeju, Korea (April 2003)
14. Collins, D.: Carrier Grade Voice over IP, 2nd edn. McGraw-Hill, New York (2002)
15. Yu, J., Choi, S.: Comparison of Modified Dual Queue and EDCA for VoIP over IEEE 802.11 WLAN. *European Trans. Telecomm.* 17(3), 371–382 (2006)
16. Cali, F., Conti, M., Gregori, E.: Dynamic Tuning of the IEEE 802.11 Protocol. *IEEE/ACM Trans. Networking* 8(6), 785–799 (2000)
17. Cai, L.X., Shen, X., Mark, J.W., Cai, L., Xiao, Y.: Voice Capacity Analysis of WLAN With Unbalanced Traffic. *IEEE Trans. Veh. Technol.* 55(3), 752–761 (2006)
18. The Network Simulator – ns-2, <http://www.isi.edu/nsnam/ns/>
19. Intersil, HFA3861B; Direct Sequence Spread Spectrum Baseband Processor (January 2000)
20. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Imielinski, Korth (eds.), vol. 353. Kluwer Academic Publishers, Dordrecht (1996)
21. Kamerman, A., Monteban, L.: WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal* 2(3), 118–133 (1997)
22. ITU-T Recommendation G.107, The E-model, a computational model for use in transmission planning (December 1998)
23. Markopoulou, A.P., Tobagi, F.A., Karam, M.J.: Assessing the Quality of Voice Communications over Internet Backbones. *IEEE/ACM Trans. Networking* 11(5), 747–760 (2003)

QShine 2009

**Invited Session IV – Mobility and QoS
Support in Heterogeneous Wireless
Mesh Networks**

Transparent and Distributed Localization of Mobile Users in Wireless Mesh Networks

Mehdi Bezahaf¹, Luigi Iannone², Marcelo Dias de Amorim¹, and Serge Fdida¹

¹ LIP6/CNRS – UPMC Univ Paris 06, France
{bezahaf, amorim, sf}@npa.lip6.fr
<http://www-npa.lip6.fr>

² Technische Universität Berlin and Deutsche Telekom Laboratories, Germany
luigi@net.t-labs.tu-berlin.de
<http://www.net.t-labs.tu-berlin.de>

Abstract. Localization of mobile users in wireless mesh networks (WMN) generally relies on some sort of flooding-based technique. Broadcasting the network is good for reliability but leads to increased latency and broadcast storm problems. This results in low efficiency of the location management mechanism in terms of packets loss and disconnection time. In this paper, we investigate a new DHT-based location management scheme through experimental evaluation on our WMN testbed. The main features of our proposed scheme are that broadcast packets are totally avoided and node localization becomes transparent to the users. We compare it to our previous flooding-based location scheme, namely EMM (Enhanced Mobility Management). Our results show improved performance both in terms of dropped packets and handover latency introduced to re-establish open sessions after a user moves.

1 Introduction

Wireless Mesh Networks (WMNs) are an emerging class of wireless networks that are able to dynamically organize and configure themselves [7]. They allow improving flexibility, efficiency, and coverage, while reducing the complexity of deployment. These characteristics make WMN an attractive solution in many scenarios, including enterprise/home networks, local/metropolitan area networks, and community networks like NYC wireless [5] and Quail Ridge Wireless Mesh Network [19]. Moreover, some industrials have already marketed WMNs [14, 6].

One of the main factors that helped the success of WMNs is its two-tier architecture, inspired from Wireless Local Area Networks (WLANs) and Mobile Ad Hoc Networks (MANET). Indeed, in WMNs there is a clear logical separation between the access subnetwork and the connectivity subnetwork. Wireless Mesh Routers (WMRs) are equipped with at least two different radio interfaces. The first interface is used at the access subnetwork providing connectivity to Wireless Mesh Clients (WMCs) in a WLAN-like fashion. The second interface, configured in ad hoc mode, is used to form a stable wireless backbone providing end-to-end connectivity between clients and with the Internet. In addition

to the architectural inspiration, WMNs have also inherited different communication principles from WLANs and MANETs, including routing protocols and localization services.

Among the various important issues to be tackled, our interest focuses on the localization of mobile WMCs. Localization service consists in maintaining information about the current position of WMCs in the system (i.e., the network topology), and rapidly updating this information with minimum overhead when WMCs move. As previously mentioned, existing localization approaches are mainly inspired from WLAN and MANET networks [8,11,22,23]. As a consequence, the natural two-tier architecture of mesh networks is not exploited and the position of WMCs must be proactively and periodically flooded throughout the network causing scalability issues [25].

In a companion paper, we proposed a localization service based on an on-demand flooding approach, namely EMM (Enhanced Mobility Management) [9]. In EMM, the localization service is separated from the routing protocol and the flooding mechanism is triggered only during a communication setup. A multicast request floods the network each time a lookup is performed. Besides the overhead generated in terms of number of messages, wireless multi-hop networks have proved to be very sensitive to flooding, resulting in very poor performance. To avoid these issues, we need a solution where both lookups and updates are done in a unicast manner.

In this paper, we propose a new distributed localization service for WMNs based on Distributed Hash Tables (DHT). DHTs have shown to be a good alternative in many research domains including domain name services, peer-to-peer file sharing system, decentralized databases, and routing protocols. Surprisingly, we have not seen any DHT-based localization service specifically designed for wireless mesh networks. By taking into account the WMN's two-tier architecture, our DHT-based approach runs only over the backbone, where there is high connectivity and links are relatively stable. In this way, WMCs have no knowledge of the DHT's existence and do not perform any DHT related operation or localization process (*transparency* principle). Each WMR owns a *slice* of the virtual space obtained by hashing the WMRs' identifier. Moreover, each WMR is the *locator* of all WMCs whose identifiers fall within its slice. To this end, we rely on an underlying routing protocol between WMRs, which greatly simplifies the management of the DHT substrate (cf., Section 3).

As a summary, the main features of our approach are:

- **Transparency:** The localization service is totally transparent to the WMCs.
- **Separation:** The localization service takes full advantage the two-tier architecture of WMNs, totally separating the roles of WMCs and WMRs.
- **Overhead Reduction:** The localization service, compared to flooding-based solutions, reduces the overhead by obtaining the positions of WMCs with a single unicast query.
- **Scalability:** The localization service operates correctly even with a large number of mobile WMCs.
- **Robustness:** The localization service avoids flooding messages throughout the network, which results in more robust communications.

The remainder of the paper is organized as follows. In the next section, we describe the motivation of our work, while overviewing EMM, our previous approach. In Section 3, we introduce our DHT-based mechanism, highlighting its main features. Then, in Section 4, we evaluate the performances of the proposed scheme. We conclude the paper in Section 5, summarizing our main results and sketching some future work.

2 Flooding-Based Location Service

While many valuable contributions have been made in order to improve the localization service of mobile users, most of them have been designed in the context of MANETs. Localizing clients in such flat networks is achieved through a routing protocol that relies on the active participation of the clients themselves.

In practice, many existing WMNs reuses solutions originally designed for flat networks, without taking into account the two-tier architecture of WMNs: both WMRs and WMCs share the same localization service, independently of their role in the network. In this case, a node's position (WMR or WMC) is limited to an entry in the routing table (examples of protocols are DSDV [27], OLSR [13], AODV [26], and BATMAN [24]). The performance of these protocols is inherently related to the routing convergence time. Moreover, the complexity of the localization service increases with known routing problems, including the continuous changes in the topology [14], the exposed and hidden terminal problems [10], and broadcast storm problems [25].

In an earlier work, we proposed EMM (Enhanced Mobility Management), a localization service based on an on-demand flooding and where the two-tier architecture of WMNs is taken into account [9]. EMM uses the same principle used in web browser to manage cookies, i.e., WMCs hold information about their latest WMR association in their NDP (Neighbor Discovery Protocol) cache. This information is given to the WMCs by the WMR, in the same way a server provides a cookie to a browser. Thus, when a WMR detects a new WMC association, it extracts from the WMC's NDP cache information about its previous attachment point, and informs the latter about the WMC's movement.

The fact that localization services of both WMRs and WMCs are based on flooding leads to the well-known broadcast storm problem [25]. To avoid the flooding issues and to have a scalable solution, some works propose *rendezvous-oriented* approaches based on a DHT mechanism in ad hoc environments [16]. Each node's unique identifier is related (through a hash function) to one or more other nodes in the system (which maintain the node's current position up-to-date). Inspired from these works, we decide to propose to adapt the same principles to the specific case of WMNs.

As EMM is a reactive localization service, no lookup infrastructure is used to maintain WMCs' current location. All location lookups are achieved by flooding the whole backbone, asking for the destination WMC's location, assuming that the WMR to which the WMC is associated with will reply. Thus, EMM makes the difference between the localization of WMRs, which is still based on a proactive

Table 1. LCTable – Local Clients Table maintains the list of WMCs locally associated to a WMR

LCTable	
WMC's IP	WMC's MAC
Local C_1 's IP	C_1 's MAC address
Local C_2 's IP	C_2 's MAC address
⋮	⋮

Table 2. FCTable – Foreign Client Table maintains the list of WMCs communicating with its local associated WMCs

FCTable	
WMCs IP	IP of WMC's WMR
Foreign C_1 's IP	IP of C_1 's WMR
Foreign C_2 's IP	IP of C_2 's WMR
⋮	⋮

routing protocol (e.g., DSDV) deployed only on WMRs, and the localization of WMCs, which is based on an on-demand flooding approach. To this end, each WMR maintains two tables to manage the communications between its local WMCs and remote WMCs. The first table, called *Local Clients Table (LCTable)*, is used by a WMR to maintain the list of locally associated WMCs (cf., Table 1). This table is automatically filled by the WMR through feedback from the wireless card driver. The second table, called *Foreign Clients Table (FCTable)*, is used by a WMR to maintain the positions of all WMCs (associated with other WMRs) communicating with its local WMCs (cf., Table 2). With such an approach, the respective WMRs of the two communicating WMCs tunnel data packets in the backbone, without the need to inject the position of each client at the WMRs along the path.

A critical operation is the lookup. In EMM, if a WMR wants to reach a specific WMC (not a local one), it sends a multicast request to all other WMRs in the backbone. The current attachment point of the destination WMC replies with a unicast packet. Obviously, the multicast request (which is in fact a broadcast) results in flooding the WMN's backbone.

When a WMC changes its attachment point, EMM uses the NDP (Neighbor Discovery Protocol) protocol [21] to discover the previous attachment point of the mobile WMC [9]. Thereby, the previous attachment point (i.e., the previous WMR) is informed about the WMC's movement. In order to recover open sessions, the new WMR sends for each WMC communicating with the local WMC a multicast message to find their positions. The same is performed by the WMR of the WMCs communicating with the local one, since the old WMR informs them that the WMC moved and a new lookup is necessary.

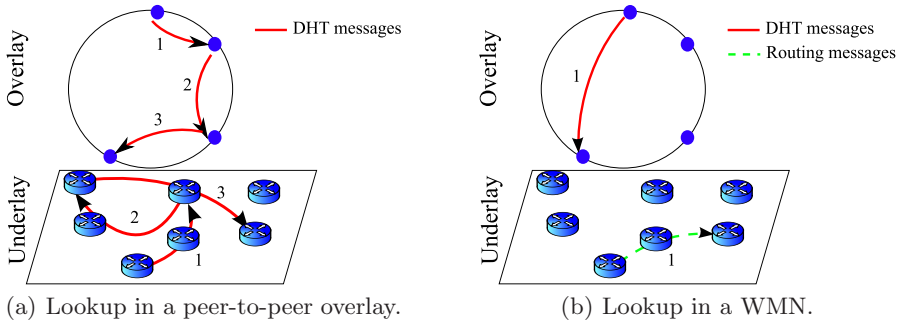


Fig. 1. DHT functionality in two different contexts. Note that in the WMN case, the system relies on an underlying routing protocol running in the backbone.

3 WMC Localization as a Shared Object in a DHT

The fundamental mechanism of a DHT consists in mapping objects' names or identifiers into keys using a hash function, such as SHA-1 [15], and distributing these keys among nodes of the system. Originally motivated by peer-to-peer file sharing systems [2,3], relying on a DHT as a location substrate also became popular in other network services, such as media streaming (audio, video), instant messaging, domain name services, and decentralized databases, to cite a few [12,20,28]). Such a success is mainly due to the attractive DHT properties: totally decentralized architecture, scalability, and robustness.

3.1 Overview of the Proposed DHT-Based Localization Service

Contrary to peer-to-peer systems, where DHT is also used to route a request concerning an object toward the server, in our proposal, the connectivity between WMRs is provided by an underlying network routing protocol that is completely independent from the DHT. The latter is only used to manage WMC's localization, mapping WMCs' identifiers into their actual locations. In an example of a file sharing system (cf., figure 1(a)), when a node wants to locate a shared file, it sends a query using the DHT, which is routed in a multi-hop fashion. However, in our WMN, when a node wants to locate a client (cf., figure 1(b)), it infers locally through the DHT the target node (solid arrow in the overlay) and then sends a request to this node through an underlying network routing protocol (dashed arrow in the underlay). This operation is detailed in the following.

The fact that our solution relies on underlying routing protocol in the backbone allows us to use any DHT model; in this work, we decided to use a DHT model based on a virtual ring \mathcal{S} inspired from Chord [29]. As the DHT runs only in the backbone, only WMRs manage slices of the addressing space.

Each node (either WMC or WMR) is assigned an identifier obtained by hashing its IPv6 address, as shown in figure 2. Let us call I_{C_i} the identifier of mesh

¹ Our implementation relies on IPv6 instead of IPv4.

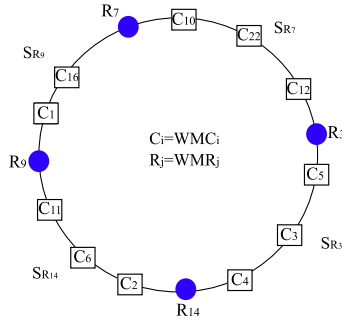


Fig. 2. DHT virtual space organization. It is a ring where the positions occupied by WMRs are indicated by circles and the positions occupied by WMCs are depicted as squares.

client C_i and I_{R_j} the identifier of mesh router R_j . The hash function we use is SHA-1, which leads to:

$$I_{C_i} = \text{SHA-1}(\text{IPv6}_{C_i}) ; I_{R_j} = \text{SHA-1}(\text{IPv6}_{R_j}). \tag{1}$$

We define $S_{R_j} \subseteq S$, the space slice containing all WMCs managed by WMR R_j . In other words, R_j is the *locator* of WMC C_i if $I_{C_i} \in S_{R_j}$. Each time a WMC changes its physical attachment point, the new WMR informs the *locator* of this WMC in a unicast fashion. Thus, to obtain the current position of any WMC, a WMR have just to contact the corresponding *locator*.

3.2 Service Architecture

In our approach, we keep the same architectural concept of EMM (cf., Section 2). We keep both the LCTable and FCTable but include two additional tables: Managed Clients Table (MCTable), which stores the list of WMCs under the control of the WMR, and Virtual Ring Table (VRTable) for the virtual ring’s maintenance.

Virtual Ring Table (VRTable): Stores the list of all WMRs participating in the DHT, in an ascending order. It allows WMRs to determine their successor and predecessor in the ring (see Table 3). Note that this is a straightforward operation. From the underlying routing protocol (which is proactive), every WMR knows all the other WMRs in the network. By applying the hash function on their IP addresses, their position on the ring is easily obtained. Since each node also knows its own position, it also knows the successor and predecessor WMNs on the ring. As a consequence, computing the slice under the control of a node is straightforward. By convention, we consider that the slice managed by a WMR is the share of the ring between its position and the position of its successor.

Table 3. VRTable – Virtual Ring Table, maintains the information about the partitioned virtual space

VRTable	
WMR's IP	Position in Virtual Ring
IPv6 $_{R_1}$	I_{R_1}
IPv6 $_{R_2}$	I_{R_2}
\vdots	\vdots

Table 4. MCTable – Managed Clients Table, maintains the list of WMCs whose identifiers are part of the managed slice

MCTable		
WMC's IP	Position in Virtual Ring	Location
IPv6 $_{C_1}$	I_{C_1}	IP of WMR with whom C_1 is associated
IPv6 $_{C_2}$	I_{C_2}	IP of WMR with whom C_2 is associated
\vdots	\vdots	\vdots

Managed Clients Table (MCTable): Maintains location information regarding the set of WMCs whose identifiers are part of the slice managed by the WMR (see Table 4). We define MCT_x as the MCTable of mesh router R_x , and $MCT_x(C_i)$ as the entry corresponding to client C_i in this table. We will see in the following how this table is filled.

For the sake of robustness, we also implement a redundancy mechanism to deal with WMRs that crash/leave the system. Contrary to P2P systems, where churn is a fundamental problem, backbones in WMN are more stable. For this reason, we assume that redundancy with three replicas is enough. Additionally to its own MCTable, each WMR maintains MCTable of both its successor and predecessor in the virtual ring.

3.3 Protocol Specification

We now present the different operations performed by a WMR w.r.t. the operation of the DHT: join, leave, lookup, and update (when a WMC moves).

WMR joining. When a WMR joins the system, it must fill up its VRTable and obtain a slice of the virtual space. We assume that, at this point, the underlying routing protocol has already performed all the operations on the routing plane (i.e., the WMR is already in the routing table of the other routers and vice-versa). The VRTable is filled with information obtained in the routing table. The node simply hashes all routers' IPs as explained in Section 3.1, obtaining the list of positions occupied in the virtual ring: $[I_{R_1}, I_{R_2}, \dots, I_{R_M}]$.

Let R_y be the mesh router joining the DHT. Computing its predecessor and successor is straightforward. They are, respectively, the routers R_x and R_z whose

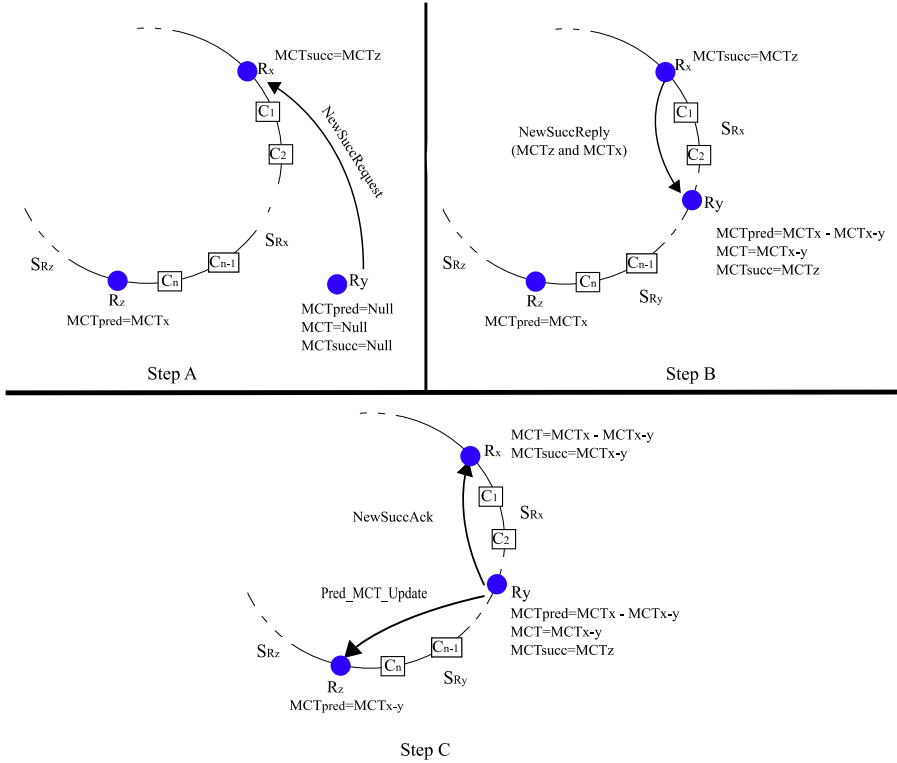


Fig. 3. WMR joining steps

identifiers I_{R_x} and I_{R_z} are immediately inferior and superior to I_{R_y} . This operation is illustrated in figure 3. R_y first informs R_x that it is its new successor in the virtual space (cf., figure 3, step A). R_x replies with both its MCTable (MCT_x) and its current successor’s MCTable (MCT_z), which allows the new router R_y to update its MCTable by processing algorithm 1 (cf., figure 3, step B) 2

Finally, R_y acknowledges R_x and notifies R_z that it is its new predecessor (cf., figure 3, step C). Router R_z replaces its previous MCT_{pred} by MCT_y (since R_y is its new predecessor).

Note that, in order for the above mechanism to properly operate, each time that the routing module detects a change in the network topology the DHT module has to be immediately notified. This allows the corresponding WMR to update the virtual ring. Also note that the notification is necessary for both joining and leaving events.

² Although the predecessor responds on behalf of its successor might result in some inconsistency problems, it saves management overhead in the system. We must however guarantee that the redundancy system works fine.

Algorithm 1. Building R_y 's MCTables.

```

for  $C_i \in MCT_x$  do
  if  $I_{C_i} \geq I_{R_y}$  then
     $MCT_{pred} \leftarrow MCT_x(C_i)$ 
  else
     $MCT_y \leftarrow MCT_x(C_i)$ 
  end if
end for
 $MCT_{succ} \leftarrow MCT_z$ 
    
```

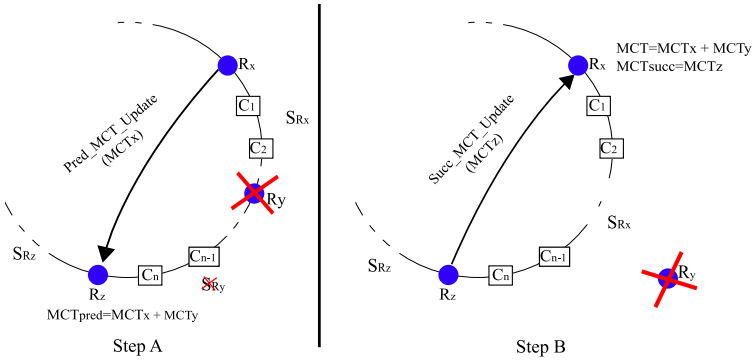


Fig. 4. WMR leaving procedure

WMR leaving. When R_x detects that one or more WMRs left (through the underlying routing protocol), it deletes from its VRTable the entry corresponding to the leaving WMR. Then, it checks locally how these changes affect its slice in the DHT virtual ring. If its successor leaves, it must send to its new successor (i.e., its previous successor's successor) its MCTable.

Its new successor locally merges the received MCT_x with its MCT_{pred} (cf., figure 4, step A). It then replies with its own MCTable. Finally, R_x merges its MCTable with its previous successor's MCTable and updates its new successor (cf., figure 4, step B).

Lookup. If a WMR needs to lookup for a specific WMC (not a local one), it starts by hashing the IP address of the WMC in order to find the corresponding *locator*. Then, it sends a unicast request asking the *locator* for the actual position of the WMC. To this end, the *locator* checks its MCTable and replies with current WMC's position (cf., figure 5). Note that this operation only involves unicast messages.

Update. When R_x detects that a new client C_i is now physically connected to its interface (cf., figure 6), it adds C_i 's IP address to its LCTable, and notifies this new association to C_i 's locator. The locator, in turn, runs algorithm 2 to check if this client is in its MCTable:

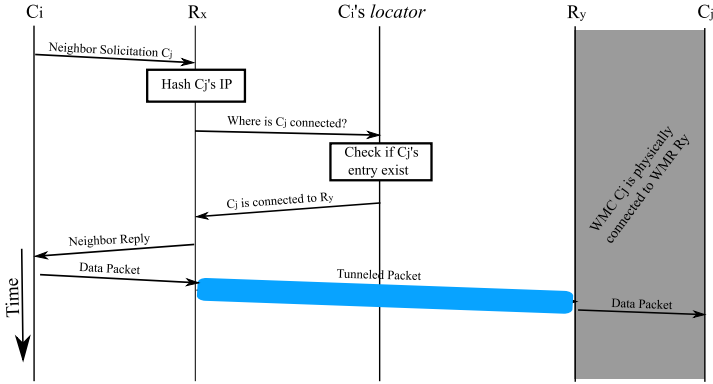


Fig. 5. WMC lookup procedure

Algorithm 2. Updating C_i 's entry at the locator's MCTable.

```

if  $C_i \in \text{MCTable}$  then
    Notify to the previous location of  $C_i$  ( $R_y$ ) that  $C_i$  has moved.
     $\text{MCTable}(C_i).\text{location} = R_x$ 's IP
else
     $\text{MCTable}(C_i) = (C_i$ 's IP,  $I_{C_i}$ ,  $R_x$ 's IP)
end if
    
```

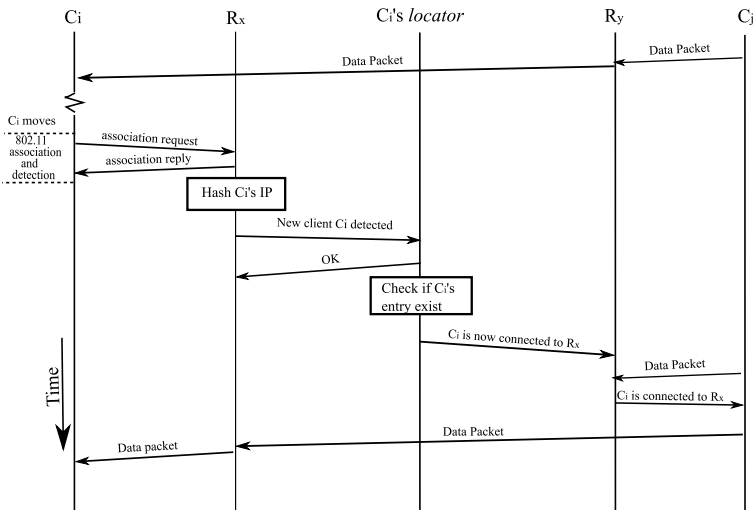


Fig. 6. WMC updating procedure

If the entry for C_i is already part of the locator's MCTable, this means that the WMC was already in the network, and it has moved to a different location. Thus, the WMR of the previous location (R_y) must be notified and the information updated. Otherwise, the locator just needs to create a new entry with the new location information. Thus, if C_i moves during communication with a WMC C_j , the WMR of the previous location (R_y) is able to forward correctly traffic to the new location.

4 Evaluation

We now present a number of results obtained through a real implementation of our DHT-based location service.

4.1 Testbed

To evaluate the performance of our approach through real deployment, we implement a Python version of our proposal, and deployed it on MeshDVNet [17]. Based on IPv6, MeshDVNet offers wireless connectivity to WMCs and allows them communicating as if they were in a traditional wireless LAN (i.e., no changes required at the WMCs). Moreover, to evaluate the efficiency, robustness, and accuracy of our DHT-based solution, we compare our results to EMM, which we recall is a flooding-based approach. We measure the performance of both EMM and our DHT on MeshDVNet using Compaq nx7000 laptops, running Linux Fedora Core 8 (kernel v2.6.24) as WMCs. All WMRs run our DHT v0.1 and Click v1.6.0 over a Linux kernel v2.6.19 on a Soekris board (AMD ElanSC520 133 MHz) with two wireless cards (802.11abg) with Madwifi driver v0.9.4 [18].

4.2 Measurement Setup

We evaluate several different mobility scenarios. We focus on the disconnection time, which represents the duration of traffic disruption that a WMC experiences during handover. To this end, we set up a test configuration where the wireless mobile client C_x changes its physical association from mesh router R_1 to mesh router R_2 while communicating with another client C_y . Three types of traffic were generated during the experiments: (i) unidirectional UDP traffic, (ii) low throughput ICMP (Internet Control Message Protocol) traffic, with its ICMP echo request messages and ICMP echo response messages (generated with the Ping command); and (iii) bidirectional TCP traffic. Concerning the UDP and TCP traffics, they have been generated and measured using the Iperf tool [30].

4.3 UDP Unidirectional Traffic

We generated UDP traffic between the C_y (UDP client) and C_x (UDP server). The generated UDP traffic is a unidirectional flow, with short messages without

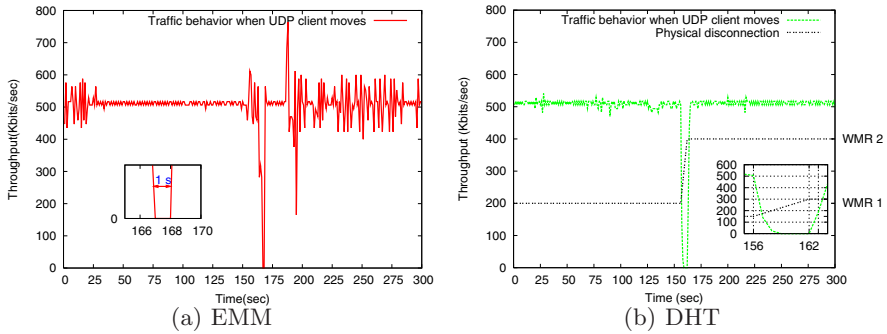


Fig. 7. Disconnection time during the handover of a UDP client

congestion control and without any arriving packet order control. Lost packets are not retransmitted.

Figure 7 shows the disconnection time when C_y moves from R_1 to R_2 . In the case of EMM (figure 7(a)), when C_y connects to R_2 , R_1 continues to receive C_x 's UDP datagrams destined to C_y . As discussed in Section 2, after C_y associates with R_2 , the latter notifies R_1 that C_y has changed its location. In turn, this allows R_1 , upon the reception of a packet destined to C_y , to notify the sender that the client has moved elsewhere. In our scenario, this means that C_x 's attachment point, after receiving this notification, proceeds to a C_y lookup to find its new location by flooding the whole backbone. R_2 replies when it receives the request, updating the location information on the WMR with whom C_x is associated. Then, C_y starts receiving datagrams again.

In our measurements, the IEEE 802.11 handover latency is practically instantaneous (a few milliseconds). We conclude then that when we use a flooding-based approach it takes around one second to recover from traffic disruption after a handover.

In the DHT approach, when C_y connects to R_2 , the latter performs a client location update by sending a unicast packet to the C_y 's locator. At the same time, similarly to the previous case, when C_y associates with R_2 , the latter notifies R_1 that C_y has changed its location, allowing R_1 to inform other routers with stale information that C_y has moved. This allows the attachment point of C_x to find C_y 's new position by sending a single unicast packet to C_y 's locator. Figure 7(b) shows the results obtained with the DHT-based approach. The figure should not be interpreted as the worst case; it happens that C_y 's card scans all 802.11's channels before connecting to R_2 . Such a complete scan lasts 6.34 seconds, while the client is not connected to any router (dashed line). From the figure, it is clear that the throughput increases instantaneously after the physical connection, meaning that, apart from the long reconnection time of the card, the traffic takes less than one second to recover.

In a second variant of the experiment, we maintain C_y static and move C_x (the server) from R_1 to R_2 . In this case, while R_2 notifies R_1 , the latter never receives

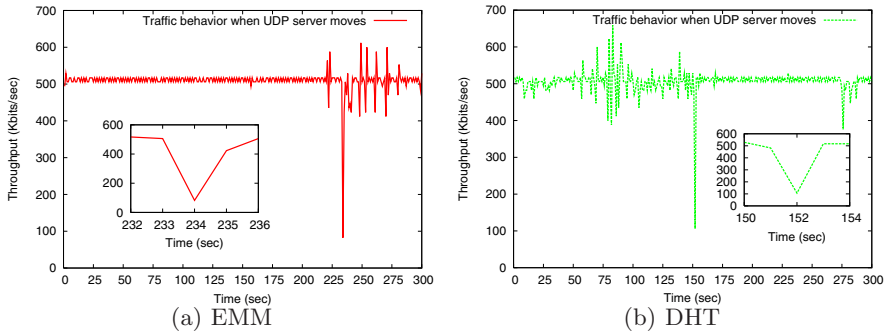


Fig. 8. Disconnection time during the handover of a UDP server

Table 5. EMM vs. DHT performance when UDP server moves

	Duration	Throughput	PDV	% of lost
Flooding	300 secs	491 Kbps	4.417 ms	4%
DHT	300 secs	505 Kbps	3.173 ms	1.3%

stale traffic for C_x , although C_x is actually sending data. What happens is that when C_x moves its new attachment point performs a client lookup in order to know where to forward the traffic destined to C_y . In the flooding approach, this means that R_2 floods the whole backbone in order to find the new location of C_y . Communication resumes when the attachment point of C_y replies. Figure 8(a) shows that throughput between second 233 and second 234 decreases from 500 Kbits/s to 82.3 Kbits/s. This means that communication is disturbed during less than one second. Between second 234 and second 235, the throughput increases from 82.3 Kbits/s to 420 Kbits/s; during this period no packets are lost.

In the DHT approach, when C_x connects to R_2 , the latter performs a WMC location update by sending unicast packet to C_x 's locator. Communication resumes when it receives the reply. Figure 8(b) shows that throughput between second 151 and second 152 decreases from 500 Kbits/s to 106 Kbits/s, which means that communication is disturbed over less than one second. Between second 152 and second 153, the throughput increases from 106 Kbits/s to 517 Kbits/s. Again, during this period no packets are lost.

The results presented above are summarized in table 5. We can observe that the DHT approach leads to good performance when compared to the flooding approach. Indeed, not only the average throughput is higher, but also the percentage of lost packet is lower and the packet delay variation (PDV) is lower, leading to reduced and stable disruption times.

The above results were obtained with a single handover per experiment. We performed as well successive server displacements (one each 20 seconds) in order to have a scenario with increased mobility. The results are shown in figure 9. As it can be seen, traffic disruption time is negligible for the DHT-based approach,

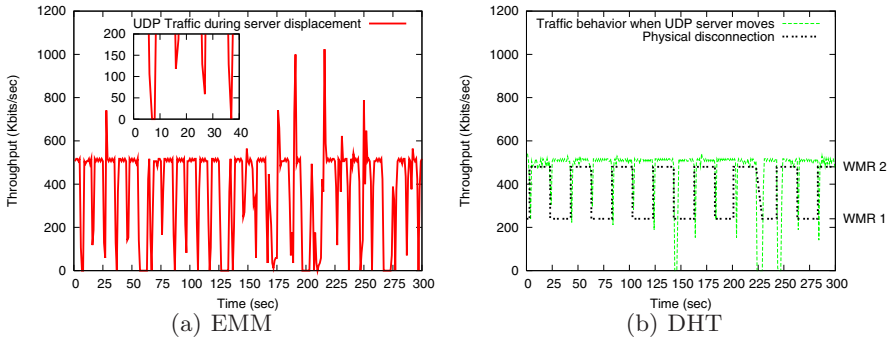


Fig. 9. Disconnection time during repeated UDP server displacements

lasting less than one second at each handover, whereas for the flooding approach traffic disruption is larger, with severe drops in the throughput. Note that in some cases the disruption time lasts for more than one second due to the driver that chooses to scan all the channels before associating the client with the new WMR.

4.4 ICMP Bidirectional Traffic

In the second set of tests, we evaluated bidirectional communications by sending ping messages between C_x and C_y , with C_x sending ICMP echo requests and C_y replying with ICMP echo responses. Packets are sent at regular interval of one second and that lost packets are not retransmitted.

Figure 10 shows the cumulative distribution function of client disconnection time at both application (Ping) and physical layers (IEEE 802.11 handover). Figure 10(a) shows what happens in the EMM case. For 78.02% of the time, the disconnection period at the physical layer is between 3 and 4 milliseconds, to which corresponds less than 3 seconds of disconnection period at application layer. The maximum disconnection period at the application layer is around 9 seconds, while the maximum disconnection period at physical layer is around 7 seconds. We conclude that using the flooding solution, updating all tables after a handover is time consuming, taking around 2 seconds.

Figure 10(b) shows the results for the DHT case. For 71.43% of the time, the disconnection period in physical layer is less than one second (between 500 and 600 milliseconds), to which corresponds less than 2 seconds of disconnection period at the application layer. Moreover, we can observe that the maximum disconnection period at the application layer is 7 seconds, while the maximum disconnection time at the physical layer is 6 seconds. This leads to the conclusion that using the DHT solution, updating all tables after a handover takes around one second. This is 50% less than in the flooding case.

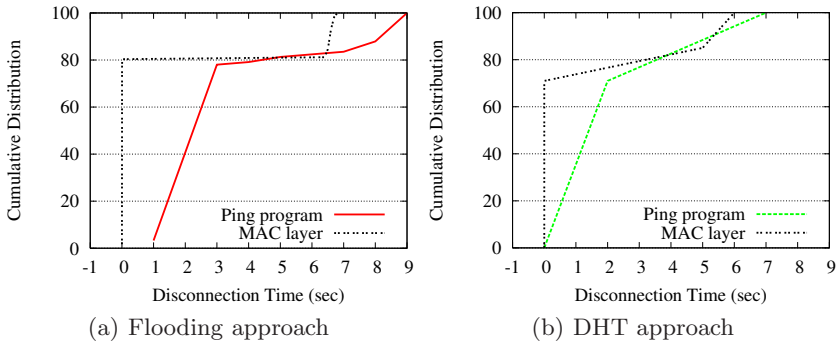


Fig. 10. Cumulative distribution function of disconnection time during handover of the ICMP sender

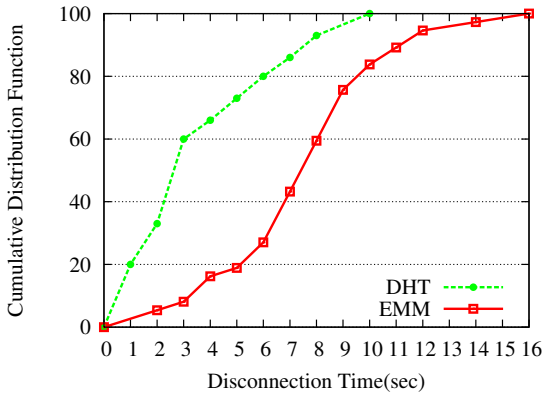


Fig. 11. Cumulative distribution function of Disconnection time during handovers of the TCP sender

4.5 TCP Bidirectional Traffic

Unlike UDP, TCP guarantees reliability and ordering of packets. This is achieved using the positive acknowledgement mechanism, which consists of retransmitting packets not acknowledged after expiration of a special timer referred as RTO (Retransmission TimeOut). In our tests, client C_x sends TCP packets to C_y . We measure the traffic disruption time in the case of successive handovers (at each 20 seconds). Unlike the tests with UDP, disconnection times during TCP tests are equal or higher than one second. The cumulative distribution function of TCP sender disconnection time during successive handovers, for both EMM and DHT, is shown in figure 11. With DHT, the disconnection time is less than 3 seconds in 60% of the cases, while using EMM it is less than 8 seconds. Moreover, we can observe that maximum disconnection time using our DHT is 10 seconds, while using EMM it is 16 seconds.

5 Conclusion

Wireless mesh networks have gained momentum in the last years and are a candidate as an enabling technology for what is known as the all-wireless Internet. Despite such a success, there are still challenging open issues. In this context, a very important problem is the lack of an effective localization service that guarantees fast and efficient mobility management.

In this paper, we proposed and evaluated through real experiments a DHT solution that relies on a proactive routing protocol running in the backbone. With our approach, we achieve not only the transparency principle but also reduced overhead thanks to the use of unicast messages only. This is orthogonal to existing solutions that rely on flooding-based mechanisms during a lookup procedure. Our results show that, for both unidirectional and bidirectional traffics, the DHT approach leads to significant performance improvements.

As future work, we intend to perform more in-vivo experiments with traffic generated by real applications and a theoretical analysis of the overhead reduction.

References

1. Cisco systems, <http://www.cisco.com/>
2. Gnutella, <http://rfc.gnutella.sourceforge.net/>
3. Napster, llc. napster, <http://www.napster.com/>
4. Nortel, <http://www.nortel.com>
5. Nyc wireless, <http://www.nycwireless.net>
6. Strixsystems, <http://www.strixsystems.com/>
7. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: a survey. *Computer Networks Journal (Elsevier)* 47(4), 445–487 (2005)
8. Amir, Y., Danilov, C., Hilsdale, M., Musăloiu-Elefteri, R., Rivera, N.: Fast handoff for seamless wireless mesh networks, pp. 83–95. ACM Press, New York (2006)
9. Bezahaf, M., Iannone, L., Fdida, S.: Enhanced mobility management in wireless mesh networks. *Journées Doctorales en Informatique et Réseaux, JDIR 2008* (January 2008)
10. Bharghavan, V.: Performance evaluation of algorithms for wireless medium access. In: *Proceedings of IEEE Performance and Dependability Symposium* (1998)
11. Capone, A., Napoli, S., Pollastro, A.: Mobimesh: An experimental platform for wireless mesh networks with mobility support. In: *Proc. of ACM QShine 2006 Workshop on Wireless mesh: moving towards applications* (August 2006)
12. Cherniack, M., Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Xing, Y., Zdonik, S.: Scalable distributed stream processing (2003)
13. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC 3626. Internet Engineering Task Force, IETF (October 2003)
14. Couto, D.S.J.D., Aguayo, D., Chambers, B.A., Morris, R.: Performance of multihop wireless networks: shortest path is not enough. In: *Proceedings of First Workshop on Hot Topics in Networks (HotNets-I)* (October 2002)
15. Eastlake, D., Jones, P.: US Secure Hash Algorithm 1 (SHA1), RFC 3174. IETF (September 2001)

16. Hubaux, J.-P., Le Boudec, J.-Y., Gross, T., Vetterli, M.: Towards Self-Organizing Mobile Ad-Hoc Networks: the Terminodes Project. *IEEE Comm. Mag.* 39(1), 118–124 (2001)
17. Iannone, L., Fdida, S.: Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks. In: *IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (RealMAN 2006)* (July 2005)
18. Madwifi Project. Madwifi – multiband atheros driver for wireless fidelity
19. Mohapatra, P., Wu, D., Gupta, D.: Quail Ridge Wireless Mesh Network: Experiences, Challenges and Findings. In: *International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities* (December 2007)
20. Motoyama, M.A., Varghese, G.: Crosstalk: scalably interconnecting instant messaging networks. In: *WOSN 2009: Proceedings of the 2nd ACM workshop on Online social networks*, pp. 61–68. ACM, New York (2009)
21. Narten, T., Nordmark, E., Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6), RFC 2461. IETF (December 1998)
22. Navda, V., Ganguly, S., Kim, K., Kashyap, A., Niculescu, D., Izmailov, R., Hong, S., Das, S.: Performance optimizations for deploying voip services in mesh networks. *IEEE Journal on Selected Areas in Communication, JSAC* (November 2006)
23. Navda, V., Kashyap, A., Das, S.: Design and evaluation of imesh: an infrastructure-mode wireless mesh network. In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)* (June 2005)
24. Neumann, A., Aichele, C., Linder, M., Wunderlich, S.: Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.), draft-openmesh-b-a-t-m-a-n-00.txt Work in Progress. Internet Engineering Task Force, IETF (March 2008)
25. Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network. In: *MobiCom 1999: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151–162 (1999)
26. Perkins, C., Belding-Royer, E., Das, S.: Ad Hoc On-Demand Distance Vector (AODV) Routing, RFC 3561. Internet Engineering Task Force, IETF (July 2003)
27. Perkins, C., Bhagwat, P.: Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In: *Proceedings of ACM SIGCOMM* (September 1994)
28. Ramasubramanian, V., Sirer, E.G.: The design and implementation of a next generation name service for the internet. *SIGCOMM Comput. Commun. Rev.* 34(4), 331–342 (2004)
29. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the ACM SIGCOMM 2001 Conference* (August 2001)
30. Tirumala, A., Qin, F., Dugan, J., Ferguson, J., Gibbs, K.: Iperf-the tcp/udp bandwidth measurement tool (2005)

Towards QoS Provisioning in a Heterogeneous Carrier-Grade Wireless Mesh Access Networks Using Unidirectional Overlay Cells

M. Kretschmer¹, C. Niephaus¹, and G. Ghinea²

¹ Fraunhofer FOKUS, Sankt Augustin, Germany
{mathias.kretschmer,christian.niephaus}@fokus.fraunhofer.de

² Brunel University, London, England
george.ghinea@brunel.ac.uk

Abstract. The visibility and success of Wireless Mesh Network (WMN) deployments has raised interest among commercial operators in this technology. Compared to traditional operator access networks WMNs have the potential to offer easier deployment and flexible self-reconfiguration at lower costs. A WMN-type architecture considered as an alternative for an operator access network must meet similar requirements such as high availability and guaranteed QoS in order to support triple-play content provisioning. In this paper we introduce an architecture of such a Carrier-grade Wireless Mesh Access Network (CG-WMAN). We then present our contribution, an approach to seamlessly integrate unidirectional broadcast cells (i.e. DVB-T) into such a CG-WMAN. This allows higher layer protocols to utilize broadcast cells like regular mesh links, where beneficial for a given payload and receiver distribution. We then present a typical use case and discuss for which combinations of traffic type, user distribution and QoS requirements the use of longer range broadcast technologies can help to improve the overall CG-WMAN performance in terms of throughput and reliability.

1 Introduction

WMNs have attracted the attention of network operators due to their increased deployment flexibility and potentially lower operational costs compared to regular rather fixed wireless operator networks. The work presented in this paper has been done within the context of the CARrier grade wireless MESH Network (CARMEN) [2] project, which aims at studying and specifying a WMN supporting carrier grade triple-play services in future heterogeneous mobile/fixed network operator environments. A CARMEN access network can complement existing access technologies by exploiting low costs mesh networking techniques. A key component of this CG-WMAN is an abstraction layer based on and extending IEEE 802.21 to allow the integration of heterogeneous wireless technologies such as IEEE 802.11, IEEE 802.16 as well as Digital Video Broadcast (DVB) and 3rd Generation Partnership Project (3GPP) technologies) in a multi-hop fashion in order to provide ubiquitous Internet access in a scalable and efficient

manner. On the control plane, the abstraction layer maps technology specific primitives onto a common set of events and commands. Upper layer modules such as self-configuration, routing, mobility management and monitoring are implemented on top of those abstract primitives and can therefore operate with any technology that provides a proper MAC adaptor. The concept of Traffic Engineering (TE) using Path Computation Elements (PCEs) as specified in [4] is adapted by the routing module to perform inter and intra area routing. Our work focuses of the seamless integration of unidirectional technologies so that they can be utilized when beneficial for a specific content or user distribution.

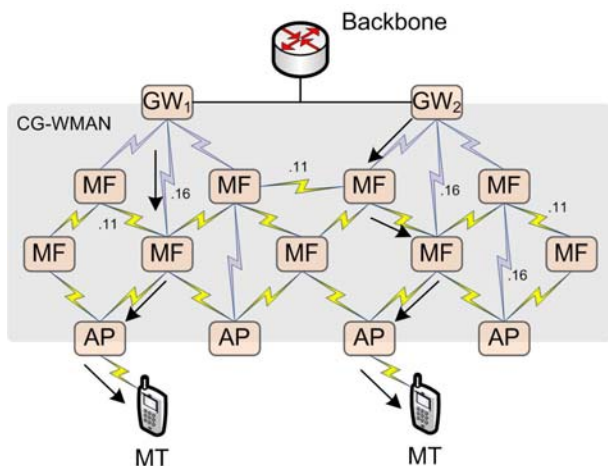


Fig. 1. Typical CG-WMAN scenario

Figure 1 depicts a typical CG-WMAN scenario with dedicated gateways, mesh forwarder as well as access point nodes and various links interconnecting them. The links might have been establish using heterogeneous technologies such as IEEE 802.11 or 802.16, chosen to optimally fit the deployment scenario with regards to financial constraints, range, spectrum availability and robustness.

Delivering triple-play services within a CG-WMAN is a challenging task since the delivery of high-bandwidth multimedia traffic substantially increases the load on the affected individual mesh links, the link groups or broadcast domains they belong to and therefore the CG-WMAN as a whole. We therefore propose the seamless integration of broadcast technologies as an efficient delivery medium especially for, but not limited to, broadcast and multicast content.

Figure 2 depicts the architecture of a CG-WMAN node. The central component of the control plane is the Media Independent Messaging Function (MIME) which can be seen as an extension of the IEEE 802.21 Media Independent Handover Function (MIHF) providing additional mesh network related primitives. It also implements a module to module communication mechanisms. Additional IEEE 802.21 compliant message types have been defined to cover this extended

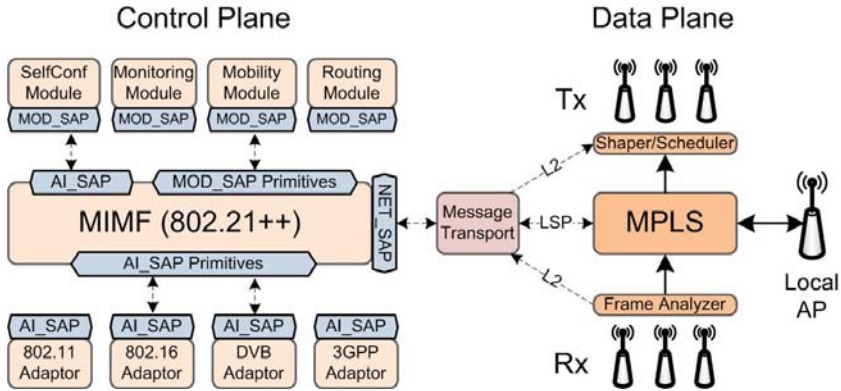


Fig. 2. The CG-WMAN Architecture is based on 802.21++ and MPLS

functionality. Messages are forwarded within the **CG-WMAN** using either a dedicated management Multi Protocol Label Switching (**MPLS**) Label-Switched Paths (**LSPs**) or via hop-by-hop data link layer forwarding. The latter can choose between controlled flooding or explicit source routing. The **CG-WMAN** internally uses EUI-64 addresses, to allow for the incorporation of non-IEEE 802 technologies. Depending on the capabilities of the technology used, the wireless interfaces are operated in a *promiscuous* mode so that all frames received by the wireless network adapter can be analyzed by the *Frame Analyzer* component. This component is crucial to the proper operation of the **CG-WMAN** since it extracts and derives information from the received frames and their headers in order to allow the monitoring component to provide this information in a standardized manner to the other **CG-WMAN** modules.

Logically below the **MIMF** component, the MAC adaptors are located. They provide the adaptation of technology specific features to the common set of primitives provided by **MIMF** to the upper layer modules via the *AI_SAP*.

The higher layer modules combined provide the functionality of a traditional routing protocol - and beyond. The functionality of a module varies depending on the function of a node. Our **CG-WMAN** design is based on a centralized approach, where the centralized management nodes maintain the state of their area following the concept of a centralized and stateful **PCF** [4]. In the centralized approach unicast or multicast path computation as well as resource accounting and allocation can be performed optimally according to **TE** policies set forth by the operator. Due to the centralized management, it is crucial for the proper operation of the **CG-WMAN** that the state maintained at the management entity closely reflects the actual state of the physical mesh links.

The forwarding component is designed based on the **MPLS** [17] specification and will be described in more detail in the next section. The so-called link group concept has been introduced to address the issue of wireless channel resources being shared by more than one transmitting node. This concept takes into account medium access protocol characteristics, the resources allocated as well as

the modulation being used in order to accurately compute and distribute the channel resources thus avoiding overbooking or contention. For infrastructure-based technologies such as IEEE 802.16 or 802.11 in managed mode, the nodes connected to the base station form a link group with the base station being the link group leader. In the case of a multicast LSP with multiple receivers in a link group, the datagrams need to be sent and accounted for with the smallest common modulation between the sending or relaying node, which is usually the base station, and the set of receivers.

The self configuration component is responsible for proper configuration of the mesh nodes, in particular it performs on-line radio planning to configure the wireless interfaces minimizing the possibility of interferences and maximizing the overall mesh network throughput. The main task is located at the management entity where it maintains a table of all possible physical links between nodes and their radios. A subset of those physical links is exported as a table of logical links which can be used for actual datagram forwarding. Physical and logical links are described as unidirectional resources. This table of logical links is similar to a link state table of traditional routing protocols such as Open Shortest Path First (OSPF) [12] and forms the basis of the TE path computation function. The details of self configuration module are out of the scope of this paper.

The mobility component provides Mobile Terminal (MT) mobility similar to the Proxy Mobile IP (PMIP) [5] concept, but is outside of the scope of this work.

1.1 Overlay Cells

Overlay cells have been studied in the literature for cellular networks where they might increase the system capacity [19] [8] [3], but also in the context of WMANs [16] [20] [11] [6]. Here mostly with the focus to break with the single-radio-per-node ad-hoc forwarding paradigm and its limitations regarding throughput and predictable Quality of Service (QoS) support. In our CG-WMAN, bidirectional overlay cells are natively supported by our architecture since to our routing module they simply appear as longer distance links between mesh nodes in the link-state table, see Figure 1. The advantage of such links, the direct link local connection between nodes needs to be balanced against the lower bandwidth per area density compared to smaller mesh cells which can exploit Space-division multiple access (SDMA) and frequency re-use, see Figure 3. Smaller cells allow for higher unicast throughput via multiple hops, while larger cells can reach a large group of receivers with a single isochronous transmission. They are therefore well suited for the distribution of multicast traffic or specific mesh network management or synchronization tasks. Hence, we propose the use of overlay cells provided via robust unidirectional broadcast technologies such as Digital Video Broadcast - Terrestrial (DVB-T), which allow for one single sending node only. Hence, complicated Media Access Control (MAC) protocols can be omitted, thus freeing up more wireless channel resources for the actual data transmission, which yields a higher physical channel utilization efficiency.

Most multicast use cases can be addressed using 1-to-N trees. Where multiple or mobile senders are required, they could be configured to send their datagrams

via unicast to the multicast tree root, which would then reflect them back out into the tree. This approach may increase the delay for some receiving nodes, but can easily be integrated into the **CG-WMAN** **QoS** management, mobility and forwarding schemes.

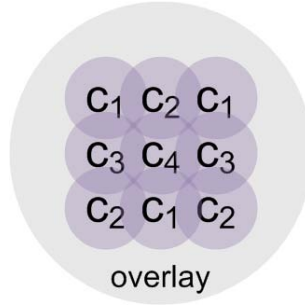


Fig. 3. Overlay cells have a higher range, but less dedicated bandwidth density

The **CG-WMAN**'s routing module must utilize proper routing metrics to match a path resource request with given **QoS** requirements to the best matching links taking into account hop-count, bandwidth, costs and modulation constraints to reach all receivers.

In a **CG-WMAN** unidirectional overlay cells might only cover parts of the mesh, therefore further in-mesh multicast forwarding might be needed to reach all receivers, see Figure 4. We therefore propose to seamlessly integrate unidirectional overlay cells into the **CG-WMAN** architecture, so that they are seen by upper layer management modules such as routing like any other technology. Then, the routing module can automatically chose between overlay cells and regular mesh links or consider regular mesh links as extension branches of the overlay cell when computing an optimal multicast tree. If supported by the underlying technology, the management modules could adapt the transmit power to control the cell size or balance between higher modulation or longer range.

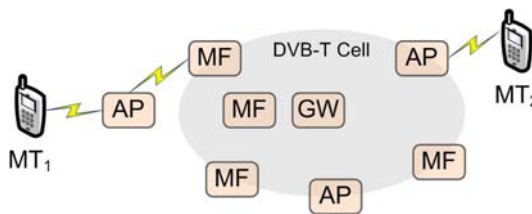


Fig. 4. A Unidirectional **DVB-T** overlay cell partially covering the **CG-WMAN**

As discussed in [10], the seamless integration of unidirectional technologies has implications on several **CG-WMAN** protocols and components, which usually

expect bidirectional links between nodes. In the following section we will address those issues and present our proposed solutions. In section three, we will analyze a typical overlay cell use case. In section four we summarize our contribution and give an outlook on ongoing and future work.

2 Approach: Integration of Unidirectional Technologies

The **CG-WMAN** described in the previous section adopted many proven **TE** concepts and protocols which are in use today in **MPLS**-based carrier backbone networks. Such networks are built on top of reliable bidirectional links and run Interior Gateway Protocols (**IGPs**) such as **OSPF** [12] or Intermediate system to intermediate system (**IS-IS**) [9]. In this section we describe the adaptations that are required in order to natively support Unidirectional Links (**UDLs**) in a more volatile **MPLS**-based **CG-WMAN**.

As discussed in [10] our **CG-WMAN** uses a fully centralized and stateful **QoS**-constrained path computation scheme based on the Dijkstra algorithm which natively considers **UDLs** if present in the link state table. Path computation can be performed for unicast paths as well as for multicast trees. Forwarding is performed along **MPLS LSPs** which are unidirectional resources and, once configured, do not require any modification.

As described in the previous section, our **CG-WMAN** does not run a link state **IGP**, mainly due to their convergence issues in the presence of volatile links. It rather relies on a centralized management entity. As a consequence, network management protocols can not assume a functional Internet Protocol (**IP**) routing between mesh nodes. Additionally, when considering **UDLs**, protocols like Address Resolution Protocol (**ARP**) or Internet Protocol, Version 6 (**IPv6**) Stateless Address Autoconfiguration (**SAA**) can not be relied on. Hence, local scope **IP** addresses would need to be configured via a specifically tailored protocol. Without an operational routing protocol, though, the use of **IP** as the signaling layer does not provide any benefit. It rather introduces extra overhead due to the additional Internet Protocol (**IPv4**) or **IPv6** header. Avoiding this overhead and addressing unidirectional MAC layer implementation, we propose to perform **CG-WMAN** messaging on the data link layer. In the centralized approach, control communication happens mainly between the management entity, which may be co-located with a gateway node, and a regular mesh node. If the management entity is not located at a gateway node, a proxy is required.

2.1 MPLS

The forwarding function supports multicast forwarding via a list of possible outputs, node-local scope labels as well as Fast Reroute (**FRR**) [13] to support fast fail-overs [15] in the case of link degradations below a pre-configured threshold. The forwarding component is designed based on the **MPLS** specification, but has been adapted to take into account **UDLs** as well as possible label collisions due to shared wireless channels where multiple sending nodes within a broadcast

domain or link group might be upstream neighbors which independently assign the same label. An upstream label negotiation, as described in RFC3031 is not possible in the presence **UDLs**. To address this issue, the forwarding component switches based on so-called Point-to-Multipoint Labels (**PMPLabels**) on the incoming side which consist of the EUI64 address of the sending interface and the actual **MPLS** label, effectively turning the **PMPLabel** into a unique 84bit label. Due to the larger label size, a label can no longer be used as an index into a table. But, even a regular 20bit label would require a table with 1048576 entries, which might not fit into the memory of a small footprint mesh node. Hence, instead our forwarding module uses a simple hash function and stores its **LSP** state in a hash map. An alternative solution would be a centralized label assignment at the management entity. This approach has not been followed since it would increase the complexity of the management function even more.

While the setup and tear down protocol is related to Resource ReSerVation Protocol - Traffic Engineering (**RSVP-TE**), it does not need to allocate link resources, since this is handled centrally at the management entity. It might however carry additional information about the **LSP**'s **QoS** requirements so that each node can configure its traffic shapers or **MAC** schedulers accordingly. This information also has to be cleared on a tear down.

Regarding **ERR**, in the current IEEE 802.21 compatible design, a Point of Local Repair (**PLR**) would subscribe to *LSP_REROUTE* events of all downstream nodes that are allowed to trigger an **ERR** action. This could lead to a storm of events since all nodes downstream of a possible link breakage will detect and signal this event at almost the same time. To address this issue, a more efficient, probably hierarchical, delivery mechanisms suppressing identical messages from downstream nodes should be investigated.

2.2 Monitoring

Monitoring is the core component of a **CG-WMAN** since it performs radio as well as frame analysis of every frame received on any of a node's interfaces. This raw information is then categorized and interpreted by one or more of the *monitoring levels*, namely the *radio level*, the *link level* and the flow or **LSP level**, see Figure 5. Information processing can be done node local or at a centralized entity which can also correlate data in order to, for example, locate sources of interference.

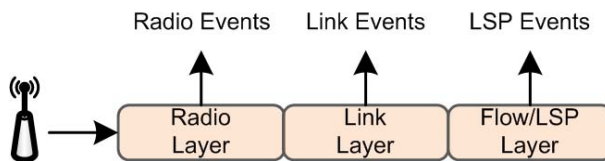


Fig. 5. The monitoring component evaluates samples at different layers

The monitoring component can create a set of events when the state of the monitored object has changed. These events trigger most actions of the other **CG-WMAN** management components, such as:

- Change of Neighboring mesh node statistics
- Mobile Terminal (**MT**) presence detection and hand-over indications
- Indication of interference or intrusion
- Change of link state, i.e. *up*, *down* or *underperforming*
- Provision of **LSP** end-to-end performance statistics
- Creation of **QoS** & network state aware **LSP** events to support i.e. **FRR**
- Network reconfiguration based on long term link analysis

Detailed wireless monitoring is a difficult task due to the dynamic nature of wireless links caused by temporary fading and interferences, but also due to often very dynamic per-frame transmitter configurations. The wireless technologies considered in our **CG-WMAN** range from satellite (i.e. **DVB-S**) over **3GPP** to **IEEE** 802.16 and 802.11. Therefore the nominal characteristics, as well as the parameters that can be analyzed may vary.

As suggested in RFC 3272 [1] our **LSP** level monitoring performs end-to-end monitoring of individual **LSP** statistics. In addition to the actual bandwidth utilization, we also maintain *PHY* status, loss, signal quality, delay and activity statistics which can indicate wireless link stability with a varying significance depending on the **QoS** requirements of the payload. This receiving side monitoring measures the actual end-to-end characteristics of an **LSP** and is therefore mandatory to verify that an **LSP** receives the agreed end-to-end **QoS** handling. The per-**LSP** **QoS** requirements are installed at each node on the path during the **LSP** setup procedure together with the **LSP** forwarding state. This monitoring approach can be implemented by passive and feedback-free analysis of received datagrams which makes it therefore also suitable for **UDL**s.

2.3 Link Layer Message Forwarding

Similar to IEEE 802.11s, forwarding mechanisms for control traffic are implemented at the data link layer. User traffic is forwarded exclusively via **LSP**s and is therefore not affected by this design decision. Since all control communication takes place among **MIME** entities, but with different forwarding requirements, we provide multiple generic message forwarding schemes to the *NET_SAP* which can then utilize the most appropriate one for each **CG-WMAN** control communication. The following messaging schemes are being provided:

- Management **LSP**
- Controlled flooding towards the destination, mainly the management entity
- Explicit source routing

Messaging via the management **LSP** is the preferred mechanism and is commonly relied upon unless the **LSP** has not yet been established or is down due to a link failure or network partitioning.

The most basic hop-by-hop data link layer forwarding scheme is implemented via a *controlled* broadcast flooding towards the destination node. In the centralized approach, this destination node is in most cases the management entity, which by itself periodically floods *mesh info* messages into the mesh network. Using this information, the controlled flooding scheme can therefore direct the flooding towards the management entity. The controlled flooding approach is related to distance vector routing as it is used by many **WMN** routing protocols such as Ad hoc On-Demand Distance Vector Routing Protocol (**AODV**) [14]. In the presence of **UDLs**, the flooding control mechanism is by-passed to ensure forwarding to the destination. This mechanism is not expected to perform highly efficient forwarding, it is rather a last resort if all other means to forward control messages fail.

If enough topology knowledge is available at the sending node, it may use an explicit source routing mechanisms which precisely describes the links to be traversed. Links are described via *LinkIDs* which consists of the EUI-64 address of the sending node's interface and the EUI-64 address of the receiving node's interface. The management entities which maintain the link state table of their area typically have all the information to calculate the source route to a destination node. It is, in fact, the same mechanism that is used to calculate paths for **LSPs**. Hence, the management entity can utilize explicit source routing themselves, or can provide other nodes with custom source routing information, for example, from node A to node D and back, see Figure 6. If **UDLs** are present, the forward and return route might not be symmetric, but it will be ensured that the return route will traverse each designated node visited by the forward route.

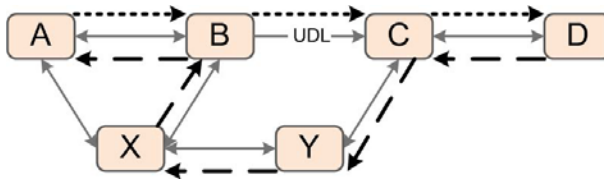


Fig. 6. The Explicit source routing scheme ensures that the same designated nodes are traversed

The explicit source routing mechanisms can operate in two different modes. In the default mode, the datagram is forwarded by intermediate nodes and only presented to the target module at the end of the route. In an alternative mode, the datagram is presented to the target module at each intermediate hop. This mode is, for example, used by the **RSVP-TE**-like path setup protocol, where state needs to be configured in each hop and is carried along the path.

2.4 Path Management Protocol

Our proposed **CG-WMAN** Path Management Protocol (**PMP**) is loosely based on the concepts of **RSVP-TE** and Path Computation Element Protocol (**PCEP**)

[18] and is realized via **MIME** messaging. **PMP** features a public interface which provides primitives for any **CG-WMAN** mesh node to

- request a path computation from a one or a set of **PCE**s
- set up an **LSP** and associated **QoS** and **MAC** layer state
- tear down an **LSP** and associated **QoS** and **MAC** layer state

PATH_REQUEST: To request the computation of a new path, this message is sent to the routing module specifying the source and destination node IDs of the path as well as the **QoS** requirements. Via the **Response** message, the routing module can return none, one or multiple possible paths to choose from. No resources are reserved at this point.

PATH_SETUP: The requesting node may then send a message to the routing function specifying the chosen path. The routing function will then try to allocate the resources and signal the setup of the associated **LSP**. Once completed, a **Response** message will be returned indicating the result of the setup procedure.

PATH_TEAR_DOWN: An established **LSP** may be torn down using this message. A **Response** message is returned when the procedure has completed indicating the result of the procedure.

Additionally, **PMP** provides a set of internal primitives which are used by the routing module to

- signal the setup of an **LSP** using explicit source routing
- signal the tear down of an **LSP** using explicit source routing
- configure **FRR** backup **LSP**s
- manage **FRR** event triggers
- retrieve **LSP** label and statistics for debugging purposes

When the routing function receives a **PATH_SETUP** message, it sends an explicitly source routed **LSP** setup message using the **MIME** link layer forwarding service which is forwarded hop-by-hop along the path to be set up. Each node will assign a local outgoing label for this new **LSP** and store this label in the setup packet, so that it can be signaled to the next downstream node as incoming label. In the presence of **UDL**s, data link layer addresses of link local neighbors can not be learned. However, the *LinkID* consists exactly of this information, the EUI-64 address of the outgoing interface and the EUI-64 address of the destination interface. A new **LSP** is identified at the ingress node using that nodes outgoing label.

A **PATH_TEAR_DOWN** message triggers a similar procedure to tear down an **LSP** identified by its ingress node's label.

For debugging purpose, a **PATH_COLLECT_STATS** message can be sent along the path of an **LSP** to collect the local labels associated with this **LSP** as well as related performance statistics.

FRR backup **LSP**s are signaled similarly to regular **LSP**s. The decision which **LSP** or segments thereof are to be protected with a backup is based on operator policies. After the backup **LSP** has been signaled, the nodes along the protected

segments can be configured with multiple specific source routes towards the **PLR**. If no specific source routes have been configured, the controlled broadcast delivery mechanism is used. In the event of a link breakage, signaling might be impacted, as well. Hence, multiple paths and forwarding mechanisms could be used in parallel.

3 Use Case: Multicast

In the previous section we have presented our approach to seamlessly integrate **UDLs** into a **CG-WMAN** so that they can be utilized when beneficial for given payload characteristics or receiver distributions. We envisage a number of different use case where **UDLs** can increase the overall network efficiency, throughput or reliability.

UDLs based on, for example, **DVB-T** offer a very robust transport, a high spectral efficiency and are not impacted by any channel access protocol overhead. Hence, they are suitable for any kind of feedback-free content delivery from broadcast multimedia content to network topology updates. The latter one can benefit from the fact that a datagram is received at all nodes isochronously. Properly implemented, even network timing or synchronization tasks could be realized.

Single source 1-to-N multicast routing within a **CG-WMAN** with overlay cells can be configured in different ways depending on operator policies, receiver distribution and **QoS** requirements:

- In-mesh hop-by-hop multicast tree forwarding
- Single hop broadcast via an overlay cell
- Using a combination of the above

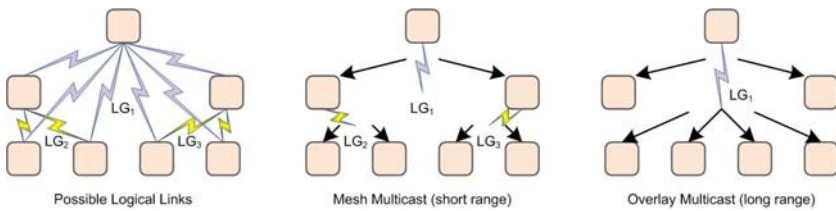


Fig. 7. Multicast routing can use in-mesh hop-by-hop forwarding or overlay cells

In order for the routing module to calculate the optimal multicast forwarding tree, a cost function is required which combines the information provided by the link group model, the link state table and the path database of established **LSPs**. From the link state table of logical links and the path database the topological receiving node distribution of a multicast tree can be determined. The result is a set of hop-by-hop forwarding trees, since in most cases multiple options

will exist to form a tree covering all nodes, see Figure 7. Taking the link group information into account, the algorithm can determine which links belong to the same link group and if the nodes could be reached with a single link local broadcast transmission and which Modulation and Coding Scheme (MCS) must be used to reach the node with the weakest link conditions in the link group. A lower MCS yields a lower spectral efficiency (E). To calculate the costs of the resources to be allocated for an LSP segment in a link group, the number of nodes (N) in this link group, the costs of its resources (C), i.e. bandwidth, and the scheduling or channel access overhead (O) are required. The latter varies heavily depending on the technology, its MAC layer design and the payload characteristics. For example, the IEEE 802.11 MAC is very inefficient when small (i.e. Voice-over-IP (VoIP)) datagrams are sent, since before sending each datagram the contention-based channel access procedure must be executed. This often requires more time than the actual datagram transmission.

Hence, the costs of LSP resources in a link group C_{LSP} can be expressed as:

$$C_{LSP} = \frac{C \cdot O}{E \cdot N}$$

C and O are constant for a given LSP and its payload's characteristics. In the example depicted in Figure 8, E has a lower bound of $E_{min} = 0.5$ bits/s/Hz and an upper bound of $E_{max} = 4.0$ bit/s/Hz and may therefore vary depending on the receiving node distribution in the link group. Since E is bound and N may raise to ∞ , C_{LSP} decreases reciprocally proportional to the number of nodes, for $N \gg \frac{E_{max}}{E_{min}}$. If N is in the order of $\frac{E_{max}}{E_{min}}$, however, adding one distant node can significantly decrease E and thus increase C_{LSP} .

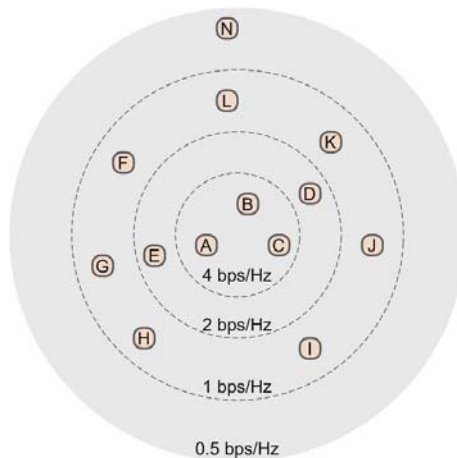


Fig. 8. The spectral efficiency decreases when a lower MCS is required to reach distant receivers or to increase reliability

For example, in Figure 8, if only node A is present, $E = E_{max}$. If nodes B and C are added, E remains constant and due to the increased number of nodes, the costs C_{LSP} decrease. Let's assume, node N should be added. In this case E would decrease significantly to E_{min} , while the number of nodes would just increase by one. It is now up to provider policies if node N would be allowed to join the link group (i.e. for premium customers), would be offered the payload via other links, or be denied access (i.e. for low-cost resale customers).

The above considerations need to be applied to each link group of the multicast tree. The total costs of a multicast tree C_{tree} can then be expressed as the sum of the costs of the n link groups traversed:

$$C_{tree} = \sum_{i=1}^n C_{LSP_i}$$

Here, we assume that the individual local link group costs consideration do not affect other links or link groups which are part of the multicast tree or even the mesh network as a whole. A simple multicast tree computation algorithm therefore has to consider two optimization criteria, meeting the end-to-end QoS requirements of the payload while minimizing the total costs C_{tree} of the resulting multicast tree.

A more complex algorithm might be required to take into account dependencies between local link group optimization, provider policies, detailed payload characteristics and estimations of usage or receiver distribution pattern variations. The above approach could serve as a base line to quantify potential optimization gains of more complex algorithms.

Summarizing the above, we have shown that for large sets of nodes overlay cells offer a viable solution to keep multicast traffic off the regular mesh links. In cases of smaller sets of nodes, multiple trade-offs need to be considered to find the optimal multicast tree.

4 Conclusion and Future Work

We have presented our proposal to seamlessly integrate unidirectional technologies at the data link layer into a CG-WMAN. We have shown that given the constraints of a CG-WMAN, this is a suitable approach. In order to optimally utilize overlay cells, we have discussed which requirements routing algorithms need to fulfill and which trade-offs to consider.

Future work will look into autonomous adjustments of link group costs according to operator policies and a more advanced multicast group computation algorithm taking into account dependencies between local link group optimization. We will also evaluate additional use cases such as the (temporary) increase of downstream bandwidth or the use of overlay cells to transmit mesh network management messages.

The work described in this paper is a work in progress. At the time of writing, the described CG-WMAN is being implemented using our C++ Simple and Extensible Network Framework (SENF) [7] and its network emulator which allows

for a mixed-mode validation of our design as well as an algorithm evaluation using real and emulated network interfaces. A multi-core Linux PC can emulate about 250 nodes, which allows us to evaluate, and optimize our proposed multicast routing algorithm.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 214994. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the CARMEN project or the European Commission.

References

- [1] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., Xiao, X.: Overview and Principles of Internet Traffic Engineering. RFC 3272 (Informational). Updated by RFC 5462 (May 2002)
- [2] Banchs, A., Bayer, N., Chieng, D., de la Oliva, A., Gloss, B., Kretschmer, M., Murphy, S., Natkaniec, M., Zdarsky, F.: Carmen: Delivering carrier grade services over wireless mesh networks. In: Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2008, September 15-18, pp. 1-6 (2008)
- [3] Deissner, J., Fettweis, G.P.: Increased capacity through hierarchical cellular structures with inter-layer reuse in an enhanced gsm radio network. *Mob. Netw. Appl.* 6(5), 471-480 (2001)
- [4] Farrel, A., Vasseur, J.-P., Ash, J.: A Path Computation Element (PCE)-Based Architecture. RFC 4655, Informational (August 2006)
- [5] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., Patil, B.: Proxy Mobile IPv6. RFC 5213 (Proposed Standard) (August 2008)
- [6] Haddad, E., Gregoire, J.-C.: Implementation issues for the deployment of a wmn with a hybrid fixed/cellular backhaul network in emergency situations. In: Proc. 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology Wireless, VITAE 2009, pp. 525-529 (2009)
- [7] <http://senf.berlios.de> (accessed 22-April-2009)
- [8] Huang, Q., Ko, K.-T., Chan, S., Iversen, V.B.: Loss performance evaluation in heterogeneous hierarchical networks. In: *Mobility 2008: Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, pp. 1-7. ACM, New York (2008)
- [9] Kompella, K., Rekhter, Y.: IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). RFC 5307 (Proposed Standard) (October 2008)
- [10] Kretschmer, M., Ghinea, G.: Seamless integration of unidirectional broadcast links into qos-constrained broadband wireless mesh access networks. In: Proc. The 4th International Conference for Internet Technology and Secured Transactions (2009)

- [11] Liu, B., Thiran, P., Towsley, D.: Capacity of a wireless ad hoc network with infrastructure. In: *MobiHoc 2007: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 239–246. ACM, New York (2007)
- [12] Moy, J.: OSPF Version 2. RFC 2328 (Standard) (April 1998)
- [13] Pan, P., Swallow, G., Atlas, A.: Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (Proposed Standard) (May 2005)
- [14] Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) (July 2003)
- [15] Raj, A., Ibe, O.C.: A survey of ip and multiprotocol label switching fast reroute schemes. *Comput. Netw.* 51(8), 1882–1907 (2007)
- [16] Reaz, A., Ramamurthi, V., Ghosal, D., Benko, J., Li, W., Dixit, S., Mukherjee, B.: Enhancing multi-hop wireless mesh networks with a ring overlay. In: *Proc. 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops SECON Workshops 2008*, pp. 1–6 (2008)
- [17] Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard) (January 2001)
- [18] Vasseur, J., Roux, J.L.: Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440 (Proposed Standard) (March 2009)
- [19] Yu, J.Y., Chong, P.H.J., Yang, M.: Performance of microcell/macrocell cellular systems with reuse partitioning. In: *Mobility 2006: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, p. 51. ACM, New York (2006)
- [20] Zhou, P., Manoj, B.S., Rao, R.: A gateway placement algorithm in wireless mesh networks. In: *WICON 2007: Proceedings of the 3rd international conference on Wireless internet, Brussels, Belgium*, pp. 1–9. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2007)

Integration of OMF-Based Testbeds in a Global-Scale Networking Facility

Giovanni Di Stasi, Stefano Avallone, and Roberto Canonico

Università di Napoli Federico II, Dipartimento di Informatica e Sistemistica,
Via Claudio 21, 80125 Naples, Italy

{giovanni.distasi,stefano.avallone,roberto.canonico}@unina.it

Abstract. PlanetLab is a global scale platform for experimentation of new networking applications in a real environment. It consists of several nodes, offered by academic institutions or companies spread all over the world, that can be shared by the networking community for its tests. The main drawback of PlanetLab is its scarce heterogeneity in terms of the access technologies it offers. In this paper we discuss the efforts we made in order to alleviate this problem. We first developed a tool that allowed us to integrate a WiFi testbed controllable by OMF (Orbit Management Framework) [16] in PlanetLab by means of a multi-homed PlanetLab node. OMF is a set of tools that make it easy to automatically execute experiments and collect measurements on a WiFi testbed. The tool we developed allows, more generally, to solve the issues that arises with multi-homed PlanetLab nodes (i.e. PlanetLab nodes having more than a network interface). In order to be able to fully exploit the potential of such PlanetLab nodes, there is the need for the users to add routing rules (e.g. rules to reach a destination through the WiFi interface, instead of the Ethernet interface). Such operation cannot be performed in a PlanetLab environment, as the rules a user adds would also affect other users' traffic. Therefore it arises the necessity of user-specific routing tables, i.e. routing tables whose rules are only valid for traffic belonging to that user. In this way the user is able to route his traffic through the WiFi interface, and make it traverse the OMF-controllable WiFi testbed, while other users' traffic continues to get routed through the default primary interface. We also had to support the integration of the OMF facilities (e.g. the OMF controller) into the user environment, which is called slice, in order to allow for the customization of the testbed (e.g. loading a specific disk image on each node) and the automatical execution of experiments. The software we developed to achieve such integration is in the process of being integrated in the code base of PlanetLab, so that anyone is able to integrate its wireless infrastructure in PlanetLab.

Keywords: heterogeneous networks, network design, experimentation, measurement, performance.

1 Introduction

In recent times various attempts to enhance the heterogeneity of PlanetLab have been made. Above all, we mention OneLab, an European Project funded by

the European Commission in its Sixth Framework Programme [2]. The project started in September 2006 with “two overarching objectives: (1) to extend the current PlanetLab infrastructure and (2) to create an autonomous PlanetLab Europe”. PlanetLab Europe is a European-wide research testbed that is linked to the global PlanetLab through a peer-to-peer federation [4]. During this project different kinds of access technologies (such as UMTS [5], WiMax [9] and WiFi) were integrated, allowing the installation of new kinds of multi-homed PlanetLab nodes (e.g. nodes with an Ethernet plus a interface). The users, however, had not been provided with a tool that allowed them to set the kernel routes required to use these new access technologies. Hence the user could not choose which interface to use to reach a given destination, limiting his experimental possibilities (e.g. the possibility to route one flow using the WiMax interface, while using the Ethernet interface for the remaining traffic). UMTS interfaces in PlanetLab did not suffer from this limitation, as the software that manages them has the ability to set for which destinations the UMTS interface is required [12]. One of the contribution of this work has been to generalize that software, allowing it work with any kind of network interface and support different operations. Another attempt to add more heterogeneity in PlanetLab is in [15]. In order to integrate an OMF-testbed in PlanetLab, the authors propose the use of a gateway PlanetLab node, whose function is to open tunnels between itself and the selected nodes in the OMF testbed. Differently from our approach, the gateway node is not a client of the OMF testbed, but creates the tunnels. A similar approach was taken in [13]. The authors aimed at integrating the VINI virtual network infrastructure [11] with OMF-based testbeds.

The rest of the paper is structured as follows. Section 2 discusses the integration of PlanetLab nodes and an OMF testbed. Section 3 provides details of our implementation, with particular emphasis on the *sliceip* tool. Section 4 illustrates the testbed we used for our experimentations, while section 5 presents the results of some experiments we carried out to test our solution. Finally, Section 6 concludes this paper.

2 Integration Scenario

The PlanetLab node in our integration scenario is a multi-homed node featuring an Ethernet and a WiFi interface. The WiFi interface is used to access the OMF testbed, which is set to work as an access network (i.e. it is connected to the Internet). The Ethernet interface is mainly used for control traffic (e.g. the traffic for accessing the node), but can also be used for experiments. The PlanetLab node is equipped with the normal PlanetLab software, with the following additions: 1) the tool we developed, called *sliceip*; 2) the *OMF NodeHandler* and 3) a locking script, i.e. a script used to lock the OMF testbed. The tool *sliceip* has the function to allow the user add slice-specific routing rules. By adding these rules, the user is able to select which access network to use for his experiment. The rules are slice-specific in the sense that they are valid only for the traffic belonging to user’s slice. The *OMF NodeHandler* is a component of OMF that

is used to setup the OMF testbed. It takes as input an experiment description and executes the required operations (e.g. loading a given disk image on a node, starting a given application on a node, etc.) by contacting the *OMF NodeHandler Agents* (simply OMF NodeAgents) installed on every node of the OMF testbed. As previously stated, the OMF testbed has to work as access network, so it has to have a node connected to the Internet and set as gateway. This gateway node, in addition to routing functions, has to do natting, as the ip addresses used in the OMF testbed are private. The OMF testbed at this stage of development can be accessed on an exclusive way. The user, in order to be able to use the OMF testbed, has to acquire a lock, by using the locking script we provide. This lock ensures that only that user can setup the OMF testbed and route his traffic through it. After acquiring the lock, the user can setup the OMF testbed by means of the *OMF NodeHandler*, and then perform the experiment. The locking mechanism is accomplished by allowing only packets belonging to the slice that locked the testbed reach the OMF control network and the WiFi interface. In future, the limitation of one user at a time will be removed¹. This will be achieved by allowing different users work on two different subsets of the OMF testbed nodes and by employing different subset of non-interfering channels.

3 Implementations Details

As we have seen in the previous section, three main components are required to enable integration between PlanetLab and OMF-based testbed. The first two, *sliceip* and the *locking script* are installed in the root context of the node (i.e. the privileged context) and have counterparts, called frontends, in the slice-context (i.e. the unprivileged user context). The frontends communicate with the tools installed in the root context by means of named pipes created by a component of PlanetLab called vsys [8]. The frontend writes in one of the two pipes, and what it has written is received by the backend, which checks the input to see if it is valid and starts the requested operation. Once the operation is completed, the backend writes in the second pipe the results, which are received by the frontend and written to the standard output for the user. The *OMF NodeHandler* does not require root privileges to run, so it can be installed by the user and run in the unprivileged slice context.

3.1 The Sliceip Tool

sliceip is the tool we developed in order to enable slice-specific routing tables in PlanetLab nodes. Using this tool, the user is able to define routing rules which apply only to traffic of his slice. This is required for the user to be able to choose which interface to use for his experiment. In particular, *sliceip* enables slice-specific routing tables by leveraging a feature of the Linux kernel and a feature of the VNET+ subsystem [6] of PlanetLab. The Linux kernel has the

¹ OMF developers are already working on removing this limitation.

ability to define up to 255 routing tables. To have some traffic routed following the rules of a particular routing table, it is necessary to associate that traffic to it by means of rules applied with *iproute2*. The rules specify packets in terms of the destination address, the netfilter mark, etc. In our case, we set the netfilter mark of packets belonging to the user's slice (i.e. the packets that are generated or are going to be received by an application running on that slice) by exploiting a feature of the VNET+ subsystem of PlanetLab. We ask this subsystem, by means of an *iptables* [\[1\]](#) rule [\[2\]](#), to set the netfilter mark equal to the id of the slice (i.e. a numeric value that identifies the slice on that node) to which they belong. We then add a rule with *iproute2* to associate the packets which belong to the slice with the routing table allocated for that slice. We also set an *iptables* SNAT rule (Source Network Address Translation) in order to set the source IP addresses of packets that are going out through a non-primary interface (the primary interface is the one the default routing rule points to). This rule is required because the source ip address of packets is set after the *first routing process* happens. In fact, in case multiple routing tables are used, the routing process follows the following steps: 1) the interface for sending the packets is selected following the rules of the main routing table and the source IP address is set accordingly (this is the first routing process); 2) if the user changes the netfilter mark of the packets in the *mangle chain* of *iptables* and a rule is defined for routing those packets with a different routing table, a *rerouting process* is triggered. This rerouting process follows the rules of the selected (i.e. the slice-specific) routing table and the interface to be used is set accordingly; 3) the packet is sent out using the selected interface. During the step 2, the source IP address is left unchanged, so we need to change it explicitly before the packets are sent during the step 3.

The user interacts with *sliceip* by means of a frontend that resides in the slice. This frontend extends the syntax of the *ip* command of the *iproute2* suite with the following two commands:

- *enable <interface>*: initialize the routing table for the user's slice, add the rule to set the netfilter mark of packets to the user's slice id, add a rule to associate those packets with the routing table of the slice and add the SNAT rule for interface;
- *disable <interface>*: remove the SNAT rule for interface, remove the rule to associate the packets of the slice to the respective routing table, and remove the rule that sets the netfilter mark of packets.

4 WILE-E Testbed

We deployed a prototype testbed called WILE-E (WireLEss Experimental) in order to show a real case of our integration strategy (see Figure [\[1\]](#)). It consists of: i) 3 Soekris net4826-48 Single Board Computers; ii) 3 business-class access point Netgear WG302Uv1; iii) 1 Linux machine acting as gateway towards the

² The *-copy-xid* PlanetLab extension to *iptables*.

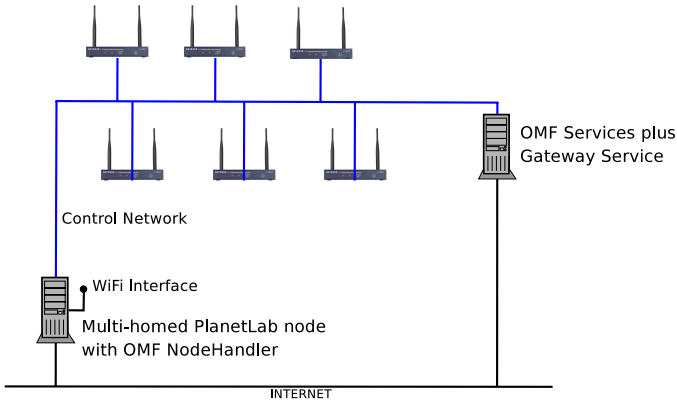


Fig. 1. WILE-E architecture

Internet for the testbed; iv) 1 PlanetLab node, through which researchers can access the Wireless Mesh testbed to run their own experiments. The PlanetLab node is equipped with two Ethernet interfaces (one with a public IP address and another connected to the internal OMF control network) and with a WiFi 802.11a/g interface to perform experiments that involve the WILE-E testbed. This PlanetLab node is associated to a private PlanetLab deployment, so we have root access for it (needed for some experiments we will show afterwards).

The Soekris net4826-50 SBC is based on an AMD Geode SC1100 CPU (at 266Mhz), has 128Mbyte DRAM memory, a 128Mbyte Flash disk, a FastEthernet interface and two 802.11a/g Atheros wireless cards. The Netgear WG302Uv1 access point is based on an Intel XScale IXP422B network processor (at 266Mhz), has 32Mbyte DRAM memory, a 16Mbyte Flash disk, a FastEthernet interface and two 802.11a/g Atheros wireless cards. We had to create two baseline images, one for each kind of device. Baseline images are the disk images the user can load on the nodes. The first image is meant to be used with the Soekris SBC. It is based on the Voyage Linux distribution [7] (v. 0.5), a distribution for embedded devices derived from Debian. This image provides two kernel images: a 2.6.20 vanilla Linux kernel and a 2.6.19 Linux kernel patched to support Click Modular Router [14]. The latter kernel can be useful for experimenters that need to run software routers constructed with the Click framework. An OMF baseline image needs to have some OMF software components which provide an interface to the OMF NodeHandler and to the OMF services for controlling the node. These components are: i) *the OMF NodeAgent*, that is the software entity, which performs local operations, such as setting the channel of the wireless interface channel, starting an application, and so on; ii) the *OMF Traffic Generator (OTG)*, that is used to generate traffic for experiments; iii) the *OMF Measurement Library (OML)*, a shared library used by OTG and, optionally, other user's application to send traffic traces to the *OML server daemon*, i.e. the OMF service whose function is to store the traffic traces. The second baseline

image is for the Netgear access points. It is based on OpenWrt [3] (Kamikaze version), a Linux distribution for embedded devices. OpenWrt supports a large number of devices and has a great community of users. In order to install the OMF components on this image, we had to create the port of some of the OMF components to OpenWrt: the NodeAgent, OTG and the OML library. We plan to submit these packages to the OpenWrt repository in order to ease for other users the process of deploying OMF testbeds that comprise nodes supported by OpenWrt.

5 Proof-of-Concept Experiments

In the following we describe some proof-of-concept experiments in order to practically demonstrate the usage of the WILE-E testbed and to characterize the behavior of the sliceip tool.

The first experiment involves the WILE-E testbed and a remote PlanetLab Europe node, located at INRIA (Figure 2). In this experiment, two end-to-end flows are generated from the multi-homed PlanetLab node that gives access to the WILE-E testbed to the remote host.

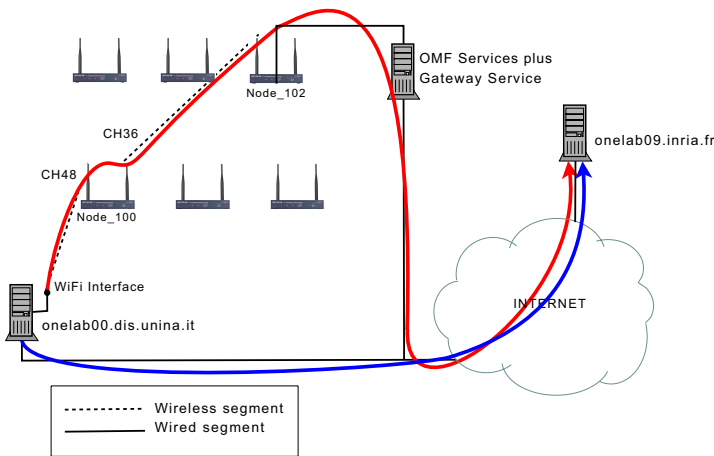


Fig. 2. An experiment with the WILE-E testbed

The experiment went through the following steps:

1. nodes onlab00.dis.unina.it and onlab09.inria.fr were added to slice
2. the OMF testbed was set up using the *OMF Nodehandler*; in particular, one of the interfaces of Node_100 was set in *Master mode* to make this node behave as an access point for the PlanetLab node;
3. a first flow was generated as a CBR flow of fixed size UDP packets; as the default routing rule of the PlanetLab node pointed to the Ethernet interface,

the packets went out through the Ethernet interface, and followed the blue path;

4. a slice-specific routing rule was added to onelab00.dis.unina.it to forward packets with destination onelab09.inria.fr through Node_100 of the Wireless Testbed;
5. a second flow was generated (with the same parameters of the previous flow); this traffic traversed the Wireless Mesh testbed and reached the Internet through the *Gateway Service* (along the red path);
6. in order to verify experiment isolation (i.e. if users of other slices were affected by the routing rule we added with `sliceip`), we generated the same flow in a second slice, and verified that this latter flow followed the blue path.

We used D-ITG (Distributed Internet Traffic Generator) [10], a platform capable to accurately generate traffic flows specified through two random processes: packet Inter Departure Time (IDT) – the time between the transmission of two consecutive packets – and Packet Size (PS) – the amount of data being transferred by the packets. Both processes are modeled as i.i.d. series of random variables, whose distribution can be selected by the user among a rich set of supported ones (constant, uniform, exponential, pareto, normal, cauchy, etc). D-ITG also incorporates some of the models proposed in the literature for the IDT and PS of the most well-known application protocols. D-ITG enables to evaluate a set of QoS performance metrics such as throughput, packet loss, delay (One Way Delay and Round Trip Time) and jitter.

The setup of the WILE-E testbed was performed by using the OMF facilities. The setup is described by the following script (interpreted by the OMF NodeHandler).

```
defGroup('gateway', [8,102]) { |node|
  node.net.w0.mode="adhoc"
  node.net.w0.type='a'
  node.net.w0.channel="36"
  node.net.w0.essid="meshnet"
  node.net.w0.ip="192.168.6.3"
  node.net.w0.netmask="255.255.255.0"
}
```

```
defGroup('ap', [8,100]) { |node|
  node.net.w0.mode="master"
  node.net.w0.type='a'
  node.net.w0.channel="48"
  node.net.w0.essid="meshnet-ap"
  node.net.w0.ip="192.168.7.1"
  node.net.w0.netmask="255.255.255.0"

  node.net.w1.mode="adhoc"
  node.net.w1.type='a'
  node.net.w1.channel="36"
```

```

node.net.wl.essid="meshnet"
node.net.wl.ip="192.168.6.2"
node.net.wl.netmask="255.255.255.0"
}

group("ap").exec('dnsmasq', ['--dhcp-range\
=192.168.7.2,192.168.7.254,255.255.255.0,infinite'])

group("ap").exec('route', ['add', '-host', \
'143.225.229.236', 'gw', '192.168.6.3', 'metric', '1000', \
'ath1'])

group("gateway").exec('ifconfig', ['eth0:1', \
'192.168.10.102'])

group("gateway").exec('route', ['add', '-net', \
'192.168.7.0', 'netmask', '255.255.255.0', 'gw', \
'192.168.6.2', 'metric', '100', 'ath0'])

group("gateway").exec('route', ['add', '-host', \
'143.225.229.236', 'gw', '192.168.10.200', 'metric', \
'1000', 'eth0'])

whenAllInstalled() { |node|
    wait 60
    allGroups.startApplications()
    wait 30
    STDIN.gets
    Experiment.done
}

```

5.1 Overhead Analysis

The aim of this experiment is to analyze the overhead introduced by our slice-specific routing mechanism.

The slice-specific routing mechanism requires, in respect to the standard routing mechanism, the execution of some extra steps: as we previously mentioned, the netfilter mark of packets has to be set to the slice-id value and different routing table (i.e. one for each slice that requests it) need to be handled.

To evaluate the overhead, we generated two end-to-end flows between two PlanetLab hosts. The source node belongs to a private PlanetLab environment (onelab00.dis.unina.it) and is equipped with two Ethernet interfaces; the sink node to PlanetLab Europe (planetlab01.dis.unina.it). The flows were two CBR (constant bitrate) TCP packet flows at 30Mbit/s and were generated both in the slice context of the source node.

Before generating the first flow, we inserted a rule in the root context, in order to make the flow go out through the second Ethernet interface. We generated the flow, collected the log file of the transmission on the sink node and removed the routing rule. Then we inserted the same routing rule in the slice context, by using sliceip. After the flow ended we collected the log on the sink node. In Figure 3, we compare the bitrate obtained for the two flows. The results for both flows are in average very close, showing that the overhead introduced is negligible.

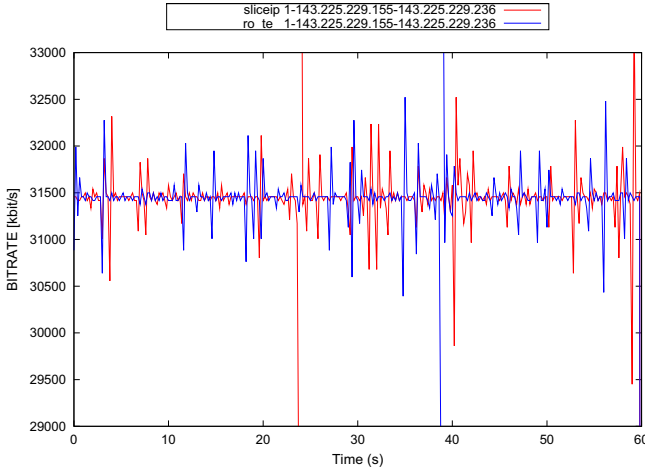


Fig. 3. Sliceip - route comparison

5.2 Redundant Routing in a Slice

This experiment shows how to exploit sliceip and the OMF WiFi testbed to perform experiments on routing redundancy with PlanetLab nodes. In the experiment we exploit the additional WiFi interface of the node to provide Internet coverage through the OMF testbed when the primary Ethernet interface becomes unavailable. This can be done by using sliceip, with the addition of a script needed to monitor the state of the Internet connection on the primary Ethernet interface. All of this is done inside the user’s slice, without affecting experiments in other slices.

The redundant routing mechanism is implemented in this way: a monitoring script continuously checks if a "test host", i.e. an Internet host chosen by the user for testing the connection, is reachable; as soon as it notices that it is not reachable anymore, the script adds a routing rule to reach the destinations selected by the user - i.e. the destinations for which the user wants to guarantee the redundant routing - through the WiFi interface. The WiFi interface is then used to reach those destinations until the script notices that the main interface is working again.

When the primary Ethernet interface fails, users with open shells on the slice will lose control of the node. This is a problem our mechanism does not address (as the control traffic is bound to the main interface). Nonetheless, the batch experiments running towards the selected destinations will continue to work.

In Figure 4 the results of an experiment of routing redundancy is shown. The graph shows the bitrate of two end-to-end transmissions between two PlanetLab hosts (the same nodes of the previous experiment). The first transmission is performed in a normal PlanetLab slice, while the second in a slice where *sliceip* is installed and the monitoring script is run. During both transmission a fault of the Ethernet interface is simulated (by shutting down the interface). As you can see, the first transmission, the one in the slice where no routing redundancy is implemented, stops working, while the second only stops for a short amount of time, after which it shows a lower bitrate. The short interruption is due to the fact that it takes some time for the script to recognize the fail of the Internet connection and some time for the new routing rule to become active. The lower bitrate is then due to the fact that the flow is traversing the OMF testbed.

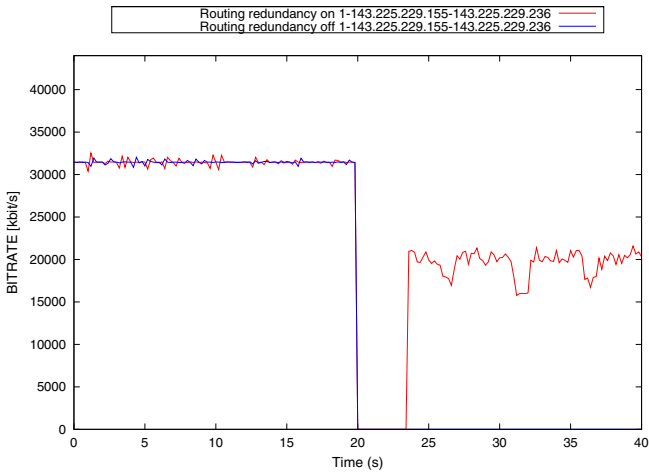


Fig. 4. Redundant routing experiment

6 Conclusions

In this paper we showed how we managed to integrate an OMF-controllable WiFi testbed in Planetlab.

This was achieved thanks to the development of two tools: *sliceip*, which allows PlanetLab users define slice-specific routing table, and the locking script, which allows PlanetLab users lock the OMF testbed for exclusive use. The main scope of the slice-specific routing table is to allow the user route his traffic through the OMF testbed, using it as an access network. It is, however, not only useful in this circumstance, but for all the cases where the user needs to control its

own routing table. We showed, on this regard, the use of sliceip to perform a redundant routing experiment.

The locking script has the scope to discipline the access of the OMF testbed among the multiple concurrent experiments running in a PlanetLab node.

Acknowledgments

Research outlined in this paper has been partially supported by the European Union under the ONELAB2 Project FP7-224263.

References

1. Netfilter/IPtables, <http://www.netfilter.org>
2. Onelab, <http://www.onelab.eu/>
3. Openwrt, <http://openwrt.org/>
4. Planetlab europe - federation, <http://www.planet-lab.eu/federation>
5. UMTS Forum, <http://www.umts-forum.org/>
6. VNET+ subsystem of PlanetLab, <http://www.cs.princeton.edu/~sapanb/vnet/>
7. Voyage Linux, <http://linux.voyage.hk/>
8. vsys, <http://www.cs.princeton.edu/sapanb/vsys/>
9. WiMax, <http://ieee802.org/16/>
10. Avallone, S., Emma, D., Pescapè, A., Ventre, G.: High performance internet traffic generators. *The Journal of Supercomputing* 35(1), 5–26 (2006)
11. Bavier, A., Feamster, N., Huang, M., Peterson, L., Rexford, J.: In VINI veritas: realistic and controlled network experimentation. In: *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 3–14. ACM, New York (2006)
12. Botta, A., Canonico, R., Stasi, G.D., Pescapè, A., Ventre, G.: Providing UMTS connectivity to PlanetLab nodes. In: *Proceedings of the 3rd International Workshop on Real Overlays & Distributed Systems, ROADS 2008, Madrid, Spain (December 2008)*
13. Hadjichristofi, G., Brender, A., Gruteser, M., Mahindra, R., Seskar, I.: A wired-wireless testbed architecture for network layer experimentation based on ORBIT and VINI. In: *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pp. 83–90. ACM, New York (2007)
14. Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.: The click modular router. *ACM Transactions on Computer Systems (TOCS)* 18(3), 263–297 (2000)
15. Mahindra, R., Bhanage, G., Hadjichristofi, G., Ganu, S., Kamat, P., Seskar, I., Raychaudhuri, D.: Integration of heterogeneous networking testbeds (2008)
16. Ott, M., Seskar, I., Siraccusa, R., Singh, M.: Orbit testbed software architecture: Supporting experiments as a service. In: *TRIDENTCOM 2005: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM 2005)*, Washington, DC, USA, pp. 136–145. IEEE Computer Society, Los Alamitos (2005)

A Proportionally Fair Centralized Scheduler Supporting Spatial Minislot Reuse for IEEE 802.16 Mesh Networks

Parag S. Mogre¹, Matthias Hollick², Jesús Díaz Gandía¹, and Ralf Steinmetz¹

¹ Technische Universität Darmstadt, Multimedia Communications Lab,
Merckstr. 25, D-64283 Darmstadt, Germany
parag.mogre@kom.tu-darmstadt.de
<http://www.kom.tu-darmstadt.de>

² Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática,
Ave. de la Universidad 30, E-28912 Leganés (Madrid), Spain

Abstract. Mesh and relay networks promise to increase the reach, capacity, and throughput of wireless communication networks. As a prominent example, the reservation-based IEEE 802.16 standard (as the basis for Worldwide Interoperability for Microwave Access WiMAX) comes with basic protocol mechanisms for an optional mesh mode as well as a relay mode of operation. This paper proposes a proportionally fair scheduler to fully utilize the potential of wireless mesh by exploiting spatial reuse. The scheduler is discussed within the setting of an IEEE 802.16 network operating with centralized scheduling in the mesh mode. We investigate the entire process of (1) bandwidth reservation, (2) calculation of the schedule and the bandwidth allocation, and (3) dissemination and activation of the schedule using an extension to the standard to allow for slot reuse. A performance analysis shows the feasibility of the proposed scheduling scheme and allows for insights into prospective future research areas in IEEE 802.16 networks.

Keywords: Wireless mesh networks, IEEE 802.16, proportional fair scheduling, spatial reuse.

1 Introduction

Networks to support wireless and mobile communications are constantly evolving towards higher data rates, scalability with respect to network coverage or number of network nodes, or improved mobility support. Extending cellular or single hop networks towards mesh or relay networks, which employ the multihop paradigm, offers a great potential for the above outlined performance improvements.

Extensions to contemporary standards for wireless communication technologies such as IEEE 802.16 ([3] [5] [6]) or 802.11 ([1] [2]) introduce the basic protocol mechanisms for mesh or relay operation. Moreover, these standards introduce the support of quality of service (QoS) using reservation-based MAC protocols. However, the aforementioned standards do only specify bandwidth

reservation mechanisms feasible for one hop operation. Moreover, they do not provide optimized algorithms and mechanisms to implement reuse-aware multi-hop scheduling.

To fill this gap, we contribute a proportional fair scheduler that supports slot reuse and is thus well suited for multihop operation. Using the example of the IEEE 802.16 standard we further demonstrate how to integrate multihop bandwidth reservation mechanisms into state-of-the-art wireless technologies, which allows us to implement and utilize our scheduler in realistic settings.

The remainder of this article is structured as follows. Section 2 surveys related work. In Section 3, the design and operation of the developed scheduler is described, and the integration of multihop reservation mechanisms in the IEEE 802.16 standard is outlined. As a proof-of-concept, in Section 4, we perform a simulation study that confirms the proper working of the developed mechanisms. The article is concluded in Section 5, where we highlight further open research issues.

2 Background and Related Work

Wireless Mesh Networks (WMNs) provide a flexible and cheap means to extend existing wireless network coverage and serve areas without existing network infrastructure. The IEEE 802.16 standard specifies a mesh mode of operation which permits the setup of WMNs able to support strict QoS requirements. To support QoS the mesh mode uses reservation based MAC protocols which explicitly reserve bandwidth for transmission for each link in the network. The mesh mode specifies two classes of mechanisms to enable the explicit reservation of bandwidth, centralized scheduling and distributed scheduling, respectively.

Using centralized scheduling nodes in the WMN can request bandwidth for transmissions for data flows to the mesh base station. The mesh base station can also allocate bandwidth for transmissions from itself to individual nodes in the network using centralized scheduling. However, these allocations are restricted to links included in a scheduling tree rooted at the mesh base station, which may cover a subset of the total nodes in the network. Distributed scheduling is more flexible and can be used for reserving bandwidth on any link in the WMN.

Using centralized scheduling, all the requests are transmitted up the scheduling tree to the mesh base station, and it then computes the allocation for individual links on the tree and generates appropriate grants. These grants are relayed down the scheduling tree to the individual nodes on the tree, which then compute the actual transmission schedules from the grant messages and using the information about their position in the scheduling tree (see also explanation in Sec. 3, and for more details readers are referred to [3] and [7]).

Centralized scheduling is thus more useful and appropriate for traffic from the nodes to the mesh base station (the mesh base station, MBS, is a node which provides access to external networks) and vice versa ([8]). In this paper

we will focus on centralized scheduling only. Centralized scheduling in the mesh mode has been investigated to some extent in the literature (e.g. [9] and [10]). However, solutions incorporating spatial reuse into centralized scheduling in the IEEE 802.16 mesh mode have not been studied to sufficient depth within the context of the IEEE 802.16 mesh mode of operation. The protocols specified in the standard for centralized scheduling do not support spatial reuse. One of the earliest works to look at use of spatial reuse within the mesh mode is [4]. In this paper the authors present an interference-aware scheduler for the mesh mode which permits central computation of an interference-aware schedule permitting multiple links to be activated simultaneously. However, here the entire schedule is determined centrally and also needs to be fully disseminated to the individual nodes else it is not possible for the nodes to find out how to schedule the actual transmissions. Further, although the authors suggest that they can use the centralized scheduling messages provided in the standard to apply their solution they do not specify any details as to how the additional reuse information will be known to individual nodes, given that the nodes are not aware of the topology of the entire wireless mesh network. Moreover, there the goal is to look towards maximizing the throughput in the wireless mesh network without considering the fairness of the bandwidth allocated to the individual links.

In this paper we investigate an extended centralized scheduler for the IEEE 802.16 mesh mode. The extended scheduler is able to schedule the centralized transmissions with spatial reuse where permissible (i.e. the same slots are used by multiple nodes where no contention would arise due to the reuse). Additionally, the allocation, and the reuse is computed by the MBS such that all nodes get a proportionally fair share of the bandwidth, and are able to fairly reuse the slots proportional to their bandwidth requirements. To the best of our knowledge, this is one of the first papers investigating extensions to the centralized scheduler in the IEEE 802.16 mesh mode to support the above goals.

3 IEEE 802.16 Reuse-Aware Proportional Fair Scheduling

This chapter presents the developed scheduler. We describe the design of the scheduler and discuss its integration into the IEEE 802.16 standard by extending it to allow for reuse-aware scheduling.

3.1 Assumptions and Requirements

For the remainder of the paper, we assume a centralized scheduling algorithm for bandwidth allocation, which is executed at the Base Station (BS) or Mesh Base Station (MBS). The scheduler operates on all bandwidth requests that are collected from the Subscriber Stations (SS) in the scheduling tree. The computed schedule, i.e., the *Minislot* allocation, is disseminated down the scheduling tree using *MSH-CSCH* messages.

Design goals for our scheduler are: proportional fairness, awareness of spatial reuse, robustness as well as good performance under heavy traffic load and scarce scheduling resources¹. We design our scheduler as follows.

- The MBS collects the bandwidth *Requests* from the SSs in a standard-conforming manner, i.e. using *MSH-CSCH* messages that are traversing up the scheduling tree in order.
- Next, the MBS computes the schedule and allocates bandwidth to the SSs in an iterative process. First, a standard-conforming and proportionally fair bandwidth allocation is determined. Next, reuse of *Minislots* is enabled by subsequent reallocation steps.
- Finally, the MBS disseminates the bandwidth *Grants* in the network. We propose an information element extending the *MSH-CSCH* control message that enables reuse of *Minislots*.

3.2 Reuse-Aware Proportional Fair Scheduling

We describe the working of our reuse-aware, proportionally fair scheduler along the operation of the centralized scheduling in the mesh mode of the IEEE 802.16 standard. We next discuss (1) the handling of bandwidth request messages and (2) the determination of the schedule.

Handling of Bandwidth Request Messages. Following the standard, the *UplinkFlow* and *Flowscale Exponent* fields of the *MSH-CSCH Request* messages are used by the SSs to indicate their bandwidth requirements (in the *Grant* message these fields and the field *DownlinkFlow* determine the granted uplink and downlink allocations, which are carried out in units of *Minislots*). The actual data rate requirement, can be calculated as follows.

$$BW_{uplink} = UplinkFlow \cdot 2^{Flowscale\ Exponent+14} \text{ bits / s}$$

Fig. 1 shows a simple multihop topology, which will serve as a sample topology to illustrate the working of the developed mechanisms. The SSs send their requests starting from the leaves of the tree. The node with the highest scheduling tree index (here SS_6) is the first SS to transmit its request. The upstream node SS_4 combines the bandwidth requirements of SS_6 with its own requirements and sends it up the tree in order, i.e., after SS_4 sends after SS_5 . As a result the MBS receives two *MSH-CSCH Request* messages containing the requests of $\{SS_6, SS_4, SS_3, SS_1\}$ and $\{SS_5, SS_2\}$, respectively.

Determination of the Schedule. The schedule is calculated in a two step process: (1) a proportional fair bandwidth/*Minislot* allocation is determined; (2) a reuse-aware policy to assign the *Minislots* is carried out. For the first step, the MBS assigns *Minislots* to nodes under the two following constraints.

¹ Note the scheduling resources are also needed for distributed scheduling in addition to centralized scheduling and should hence be used efficiently by the individual schedulers.

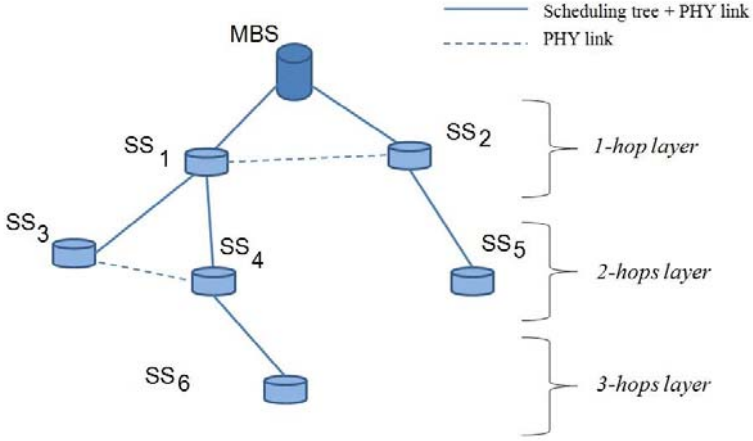


Fig. 1. PHY links and centralized scheduling tree of sample topology

- *Minislots* are allocated to single nodes and conforming to the IEEE 802.16 standard. This first allocation allows no *Minislot* reuse.
- The SSs are served in order and each SS receives a block of consecutive *Minislots*, the amount of which is proportionally fair to the total amount of requested bandwidth.

This initial assignment prevents SSs from starvation, and allows neglecting interference issues. Please note that the achievable fairness depends on the configuration of the number of permissible *Minislots* in the data subframe. The actual calculation of the schedule is straight forward. The MBS builds a table ordered according to the node indices and extracts the bandwidth requests from the different *MSH-CSCH* messages as shown in Table 1 for our example.

Next the MBS iterates over the nodes and carries out two calculations. First, guaranteeing fairness requires the evaluation of each bandwidth request in relation to the total bandwidth requested. The fair fraction of bandwidth BW_{alloc_i} for SS_i is calculated as:

$$BW_{alloc_i} = \frac{BW_{req_i} \cdot BW_{total}}{\sum BW_{req_j}} \tag{1}$$

Where BW_{req_i} represents the bandwidth requirement of node SS_i , BW_{total} is the overall amount of bandwidth to be allocated in this new schedule and the sum over BW_{req_j} describes the bandwidth requested by all nodes except SS_i . Second the actual calculation of the corresponding amount of *Minislots* for each SS is performed:

$$Minislots_i = \left\lceil \frac{BW_{alloc_i}}{BW_{minislot}} \right\rceil \tag{2}$$

Table 1. Example of the bandwidth requests propagated up the scheduling tree towards the BS. The higher the tree index, the earlier the corresponding *MSH-CSCH Request* is transmitted to allow for aggregation of requests.

Node: <i>tree index</i>	Bandwidth request	Bandwidth allocation
$SS_1 : 01$	$BWreq_1 + BWreq_3 + BWreq_4 + BWreq_6$	$BWalloc_1(Minislots_1)$
$SS_2 : 02$	$BWreq_2 + BWreq_5$	$BWalloc_2(Minislots_2)$
$SS_3 : 03$	$BWreq :_3$	$BWalloc_3(Minislots_3)$
$SS_4 : 04$	$BWreq_4 + BWreq_6$	$BWalloc_4(Minislots_4)$
$SS_5 : 05$	$BWreq_5$	$BWalloc_5(Minislots_5)$
$SS_6 : 06$	$BWreq_6$	$BWalloc_6(Minislots_6)$
Sum	Sum of all above $BWreq$	$BWtotal$ (Total amount of <i>Minislots</i>)

Where $BW_{minislot}$ is the number of bits that can be transmitted within one *Minislot*. Following this assignment, no node is left without *Minislot*; if the amount of data is lower than a *Minislot* payload, a single *Minislot* is allocated. The resulting allocation is illustrated in Table 1. It is important to notice that owing to the rounding up of bandwidth for the *Minislot* distribution the node served last could be assigned less *Minislots* than would be proportionally fair. We consider this last node first in our reallocation of *Minislots* to account for this.

The outlined scheme starts allocating bandwidth with the leave nodes of the scheduling tree, since for uplink traffic, the appropriate serving order is from the leave to the root of the tree. Related work discusses alternate schemes, though.

3.3 Reuse-Aware Scheduling for IEEE 802.16

As long as no reuse is intended, the implementation of the determined schedule in IEEE 802.16, i.e. the dissemination of the corresponding *MSH-CSCH Grant* messages, can follow the standard procedure. However, if we plan to allocate *Minislots* multiple times to non-interfering links, we cannot easily utilize the existing *MSH-CSCH Grant* messages, because currently the standard does not foresee reuse. Moreover, since the schedule is calculated in a distributed fashion based on the amount of *Minislots* assigned by the BS and disseminated in the *MSH-CSCH Grant*, we have to ensure that the SSs do not derive an interfering schedule if we add reuse information.

After the initial assignment of slots to the SSs, we propose to assign additional slots for reuse to the SSs that are to be utilized only if the interference constraints are fulfilled. During this second allocation, the nodes are served in the order starting from the lowest tree index, i.e. starting from the root of the tree. The algorithm to allocate *Minislots* for reuse operates as follows (Fig. 2 illustrates the reuse of *Minislots*):

- A list containing all SSs is created, $SS = SS_i$.
- For each block of allocated *Minislots* from the first round ($SS_x, mBlock_x$), a SS_y (if exists) that does not violate the interference constraints with SS_x and

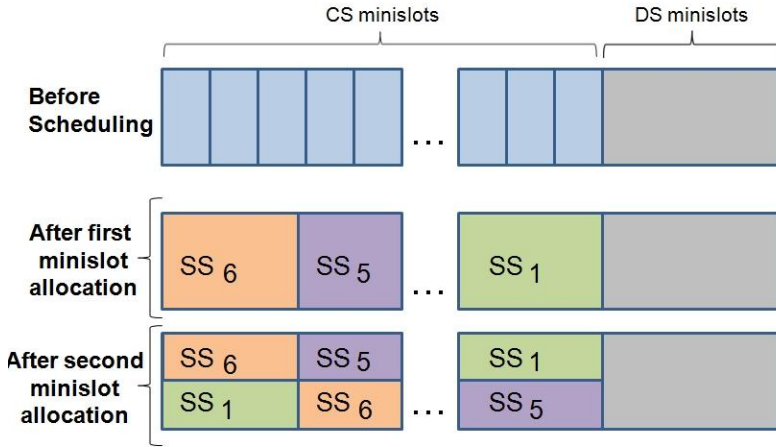


Fig. 2. Schematical operation of the reuse-aware *Minislot* allocation strategy. The example shows the reuse of *Minislot* among non-interfering links that permit for reuse (here $\{SS_6, SS_1\}$, $\{SS_5, SS_6\}$ and $\{SS_1, SS_5\}$ are node-pairs with non-interfering links).

is not already active during this allocation is selected, and another allocation ($SS_y, mBlock_x$) is performed.

The actual result of the reallocation process depends on the topology of the network, which determines the interference and reuse constraints. The obtained reuse schedule needs to be disseminated in the network. For this, it is necessary to modify the way the *MSH-CSCH* messages handle the granting process to the SSs, in order to allow for reuse. We propose to add a *Minislot* start and *Minislot* duration for the transmission (a similar mechanism is in use for the distributed scheduling in IEEE 802.16), to unambiguously indicate the *Minislot* blocks to be reused. Table 2 shows the proposed message format of the *MSH-CSCH Information Element*.

Table 2. Proposed MSH-CSCH Grant Information Element to extend the IEEE 802.16 Standard

Syntax	Size	Notes
MSH-CSCH_Grant_Info(){		
LINK ID	8 bit	
Start Frame number	4 bit	
Minislot start	8 bit	
Minislot range	8 bit	
Direction	1 bit	
Persistence	3 bit	
}		

After receiving a *MSH-CSCH Grant* message, a node processes the Information Elements that are significant for itself and retransmits the entire *MSH-CSCH Grant* message to its children. With the information included in the modified *Grant* message (see Table 2), all nodes are informed on exactly when they are allowed to transmit data for the reused *Minislots*, thus avoiding collisions due to interference.

In contrast, in the original standard the nodes are able to determine the order of transmission using the knowledge of the bandwidth allocation in combination with their own tree index, which is sufficient to determine the order of transmission only if no reuse is permitted.

3.4 Summary

The developed strategy allows for a *Minislot* allocation with a high level of fairness, while being simple and efficient at the same time. Although the described scheme does not support traffic class differentiation when making the bandwidth assignments, basic priority policies could be implemented at local level (at the individual SSs) when using the *Minislots*, by serving first to those flows which require less delay. However, thanks to the robustness of the centralized scheduler, the fairness of the proportional allocation and the performance gains of the *Minislots* reuse, a good performance results in terms of bandwidth use and latency can be expected if the network is not overloaded.

4 Proof-of-Concept in an IEEE 802.16 Mesh Network

We implemented the designed scheduler as well as the modified IEEE 802.16 protocol messages in a standard compliant IEEE 802.16 simulation environment. We utilized the mesh mode of the standard in combination with centralized scheduling. Goals of the simulation study were to confirm the proper operation of the scheduler as well as getting an estimate on the achievable performance gains over a baseline non-reuse-aware scheduler.

We use the topology shown in Fig. 1 for our analysis, which allows for reuse, but also imposes interference constraints between various branches and sub-branches of the scheduling tree. We have studied various sets of workload ranging from low to very high offered traffic load between the MBS and the individual SSs (the traffic being uniformly distributed among the SSs and directed to the MBS and vice versa); the simulation parameters are given in Table 3.

In Fig. 3 and Fig. 4, we show the results for the average delay per data packet obtained under medium and high traffic load. We have chosen the delay metric, since it is an indicator of both the network performance and the proper reuse of resources; for the same amount of admitted traffic, a lower delay indicates that *Minislots* or Frames earlier in time can be utilized, for a highly loaded network the increased throughput of the network is hindering the build-up of queues for a reuse-aware scheduling scheme.

The results shown in Fig. 3 and Fig. 4 indicate that our scheduler meets the design goals, which is particularly evident for the high traffic setting shown in

Table 3. Simulation parameters and settings for the simulation study

Parameter	Setting/Range
PHY	WirelessMAN-OFDM ($N_{FFT} = 256$), <i>ETSI</i>
Channel bandwidth	14 MHz
Subcarrier	Spacing: 62,5 kHz, 192 subcarriers
Symbol time	Overall: 18 μ s, without preamble/guard: 16 μ s
Frame duration	20ms
<i>Minislot</i> duration	1111symbols
Control subframe	12 Transmission opportunitites each with 4 symbols Modulation: QPSK-1/2
Data subframe	255 <i>Minislots</i> each with 4 symbols 1 <i>Minislot</i> with 7 symbols Modulation: 16-QAM-1/2

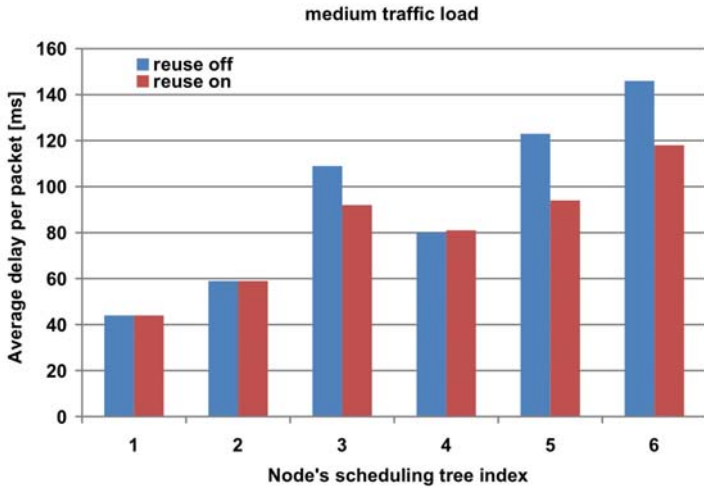


Fig. 3. Average end-to-end delay per node for reuse-aware proportional fair scheduling vs. reuse-unaware proportional fair scheduling under low traffic load

Fig. 4: the offered load can no longer be sustained for the deeper levels of the tree topology by the reuse-unaware scheme, which can be seen in the more than linear increase in delay on the two and three hop paths.

In Fig. 5 we show the differences between nodes on the same layer/tier of the topology using our algorithm. Our proportional scheduler allocates more bandwidth to nodes with more children to accommodate the potentially higher bandwidth requests of these branches of the tree. The average delay for different offered traffic loads is again an indicator for the achieved fairness of the bandwidth allocation.

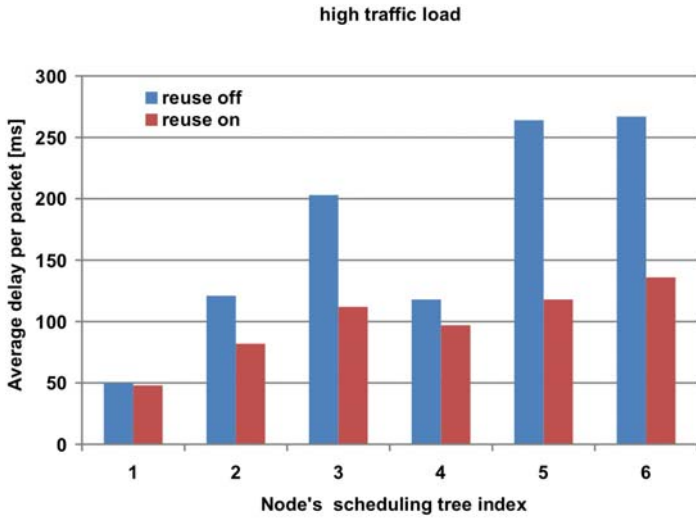


Fig. 4. Average end-to-end delay per node for reuse-aware proportional fair scheduling vs. reuse-unaware proportional fair scheduling under high traffic load

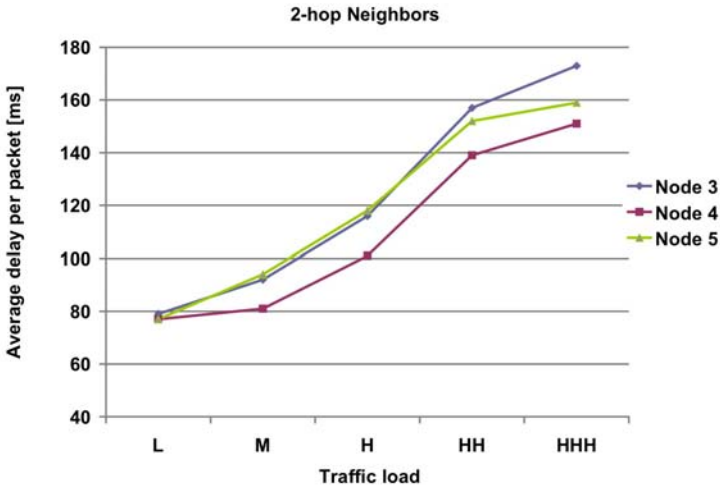


Fig. 5. Average end-to-end delay for the two hop neighbors of the MBS for reuse-aware proportional fair scheduling under varying traffic load from low (L) to very high (HHH)

We observe that our scheduler shows only small differences in delay between the nodes on the same tier of the topology. This is not the case for the reuse-unaware schemes as can be seen in Fig. 3 and Fig. 4, where we observe significantly different delays for e.g. nodes SS_3 , SS_4 , SS_5 which are on the same level of the scheduling tree.

5 Conclusion

We have proposed a proportionally fair scheduler to fully utilize the potential of wireless mesh and relay networks by exploiting spatial reuse. The developed scheduler can be easily be integrated with contemporary wireless technologies such as IEEE 802.16 that are following a deterministic and reservation-based MAC protocol. In the context of IEEE 802.16, our scheduler allows to maintain the basic protocol mechanisms for requesting bandwidth. The scheduling and bandwidth allocation to permit reuse have been designed and implemented. Moreover, extensions to the standard (that does not sufficiently support bandwidth reuse in its current specification) have been proposed to facilitate the dissemination of the derived schedule in a reuse supporting manner.

A performance analysis has demonstrated the feasibility of the developed schemes. The obtained results are promising and indicate significant performance gains, even in basic network topologies. Still, more advanced scheduling schemes and strategies can be foreseen, thus fully utilize the potential that has been opened up with the outlined the standard extension that permits for *Minislot* reuse in IEEE 802.16 mesh networks.

References

1. IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE (June 2007)
2. IEEE P802.11s/D1.08. Draft Amendment to Standard IEEE 802.11: ESS Mesh Networking. IEEE (January 2008)
3. 802.16 IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems (2004)
4. Wei, H., Ganguly, S., Izmailov, R., Haas, Z.J.: Interference-Aware IEEE 802.16 WiMax Mesh Networks. In: Proceedings of 61st IEEE Semiannual Vehicular Technology Conference VTC Spring, Stockholm, Sweden (2005)
5. 802.16 IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands (2006)
6. 802.16 IEEE Relay Taskgroup: P802.16j Baseline Document - Multihop Relay Specification. 2006. 802.16j-06/026r3, <http://wirelessman.org/pubs/80216j.html>
7. Mogre, P.S., Hollick, M., Steinmetz, R.: The IEEE 802.16 MeSH Mode Explained, Technical Report, KOM, TU Darmstadt, <ftp.kom.tu-darmstadt.de/pub/TR/KOM-TR-2006-08.pdf>
8. Redana, S., Lott, M.: Performance Analysis of IEEE 802.16a in Mesh Operation. In: Proceedings of the 13th IST Summit, Lyon, France (2004)
9. Schwingenschlögl, C., Mogre, P.S., Hollick, M., Dastis, V., Steinmetz, R.: Performance Analysis of the Real-time Capabilities of Coordinated Centralized Scheduling in 802.16 Mesh Mode. In: Proceedings of 63th IEEE Semiannual Vehicular Technology Conference VTC Spring, Melbourne, Australia (2006)
10. Du, P., Jia, W., Huang, L., Lu, W.: Centralized Scheduling and Channel Assignment in Multi-Channel Single-Transceiver WiMax Mesh Network. In: Proceedings of IEEE Wireless Communications and Networking Conference WCNC, Hong Kong (2007)

QShine 2009

**Invited Session V – Data and
Information Processing and
Management in Sensor Networks**

Cooperative Training in Wireless Sensor and Actor Networks

Francesco Betti Sorbelli¹, Roberto Ciotti¹, Alfredo Navarra¹,
Cristina M. Pinotti¹, and Vlady Ravelomanana²

¹ Department of Computer Science and Mathematics, University of Perugia, Italy
{navarra,pinotti}@dmi.unipg.it

² Laboratoire d'Informatique de Paris-Nord, University of Paris, France
vlad@lipn.univ-paris13.fr

Abstract. Exploiting features of high density wireless sensor networks represents a challenging issue. In this work, the training of a sensor network which consists of anonymous and asynchronous sensors, randomly and massively distributed in a circular area around a more powerful device, called actor, is considered. The aim is to partition the network area in concentric coronas and sectors, centered at the actor, and to bring each sensor autonomously to learn to which corona and sector belongs. The new protocol, called *Cooperative*, is the fastest training algorithm for asynchronous sensors, and it matches the running time of the fastest known training algorithm for synchronous sensors. Moreover, to be trained, each sensor stays awake only a constant number of time slots, independent of the network size, consuming very limited energy. The performances of the new protocol, measured as the number of trained sensors, the accuracy of the achieved localization, and the consumed energy, are also experimentally tested under different network density scenarios.

Keywords: wireless sensor network, training, localization, distributed algorithms.

1 Introduction

Miniaturized, low-cost, battery-operated nodes, which integrate sensing abilities, signal processing and wireless communication are well known as *sensors*. In this work, Wireless Sensor and Actor Networks (WSAN) are considered, which consist of massively and randomly deployed sensors plus few more powerful entities, called *actors*.

The random deployment results in sensors initially unaware of their spatial coordinates. Since the sensed data is of scarce utility unless related to the localization of the sensors that collect them, each actor organizes the sensors in its range of transmission (the so called *actor-zone*) in a dynamic virtual infrastructure which provides the sensors with a coarse-grained localization awareness. Specifically, the actor arranges its zone into equiangular sectors and equiwidth

concentric coronas centered at the actor itself, imposing a discretized polar coordinate system. In doing so, the actor-zone is subdivided into small regions, one for each corona-sector intersection.

The task that allows each sensor in the actor-zone to acquire its corona (sector, resp.) coordinate is known, in the literature, as the *corona (sector, resp.) training* process. The new protocol, called *cooperative*, is analytically studied under the assumption that the density of the random distributed network is sufficiently high to guarantee that each sensor is trained. The new protocol is the fastest training algorithm for asynchronous sensors, and it matches the running time of the fastest known training algorithm for synchronous sensors. Moreover, during the training, each sensor stays awake only a constant number of time slots, independent of the network size, saving thus energy.

The remainder of this paper is organized as follows. Section 2 defines the network model. Section 3 presents the *cooperative* training algorithm by specifying the actor and sensor behaviors. Assuming the network to be sufficiently dense to train all the sensors, Section 4 studies the algorithm performances measured in overall running time, drained energy per sensor, number of trained sensors and accuracy of the achieved localization. Section 5 experimentally validates the results in Section 4, and argues on the actual network density needed to train all the sensors. Finally, Section 6 offers concluding remarks and open problems.

2 The Network Model

In this section, network model and assumptions are described. At first, the virtual coordinate system to be established in the network is as follows:

1. *Coronas*: The actor-zone area is divided into k coronas C_0, C_1, \dots, C_{k-1} of fixed width $\rho > 0$, centered at the actor, determined by k concentric circles whose radii are $\rho, 2\rho, \dots, k\rho$, respectively;
2. *Sectors*: The actor-zone area is divided into h equiangular sectors S_0, S_1, \dots, S_{h-1} , originated at the actor, each having a width of $\frac{2\pi}{h}$ radians.

The actor is equipped with a long-range radio and an isotropic antenna and it is able to broadcast with variable-range R in order to reach all the sensors at distance at most $R \leq k\rho$.

The time is ruled into slots, with sensors and actor using in-phase and equally long slots. Nonetheless, since the *asynchronous* model is adopted, the sensors are not engaged in any explicit synchronization protocol and each sensor starts to count the time from when it wakes up for the first time. Thus, the same time slot corresponds to different local times for sensors which woke up at different times. The time slot when the training process starts is numbered 0 at the actor. From now on, the time slot numbering done at the actor is called *global* time, whereas that at each sensor is indicated as *local* time.

Each sensor is *anonymous*, that is, it has no individual unique ID and works *unattended*. A sensor is called of *type* x , with $x \in [0, k - 1]$, if it wakes up for

the first time at the global time x . However, each sensor is only aware of its own local time, and it has no idea of the global time. Each sensor alternates between *awake* and *sleep* periods. The sensor awake-sleep cycle has a total length of L time slots, out of which each sensor is awake for d slots and in sleep mode for $L - d$ slots. The i -th, with $i \geq 1$, awake-sleep period of a sensor of type x starts and finishes at the global time slots $x + (i - 1)L$ and $x + iL - 1$, respectively. In order to save energy, a sensor which is not required to be active in an awake-sleep cycle can skip it staying in sleep mode for L time slots.

The sensors are equipped with a small-range radio and an isotropic antenna, and during an awake period they can transmit or listen to either the actor or the sensor neighbors. A sensor can transmit in two modalities: with transmission range equal to $r = \rho$ for routing purposes or equal to $r < \rho/2$ for the cooperative training algorithm. If an awake sensor receives more than one message at the same time, we assume that it correctly receives the message only if all the transmissions refer to the same message. Otherwise, the sensor hears *noise*.

3 The Cooperative Corona Training Algorithm

In this section we present the cooperative training algorithm, which localizes each individual sensor in the actor-zone. From now on, we will assume the corona width $\rho = 1$ and the awake-sleep period $L = k$.

The cooperative training consists of three stages: the first stage is deterministic and it is the only one that involves the actor. Immediately after deployment, the actor starts to transmit. Let $|a|_k$ denote the *modulo* operation, that is the nonnegative remainder of the division of a by k . At time slot t , with $0 \leq t \leq k + d - 2$, it transmits the beacon $|k - 1 - t|_k$ at a power level sufficient to reach all the sensors up to corona $C_{|k-1-t|_k}$, but not those beyond $C_{|k-1-t|_k}$.

For sensors, the protocol has three stages. The first stage deterministically trains a certain percentage of sensors. In the other two stages, in contrast, the percentage of sensors trained strictly depends on the network density.

Each sensor has its own local time τ , which is set to 0 when the sensor wakes up for the first time, and a counter j of the awake-sleep cycles passed from the beginning of the training protocol.

The pseudo-code for the sensor protocol is given in Appendix Figure 6. The first stage lasts one awake-sleep cycle for each sensor. The sensors alternate an awake period of d time slots with a sleep period of $L - d = k - d$ time slots. At time slot t of the awake period, with $0 \leq t \leq d - 1$, each sensor listens to the actor and stores in $C[t]$ either the beacon received by the actor or the mark \emptyset when no beacon is received. A sensor in corona γ which is awake while the actor transmits beacon b receives such a beacon if and only if $b \geq \gamma$. A sensor becomes trained by the actor, and hence it becomes a *seed*, when one of the two following *Training Conditions* is verified.

TC 1: A sensor residing in corona 0 receives beacon 0. In fact, only sensors inside corona 0 can receive such a beacon.

TC 2: A sensor residing in corona γ receives beacon γ but not beacon $\gamma - 1$ when it knows that the actor is transmitting beacon $\gamma - 1$.

Since the above training condition TC2 can only be verified if $d \geq 2$, from now on, we assume $2 \leq d < k$.

In the second stage of the corona training protocol, the sensors communicate among them in order to broadcast the corona identity from the seeds to the untrained sensors. In other words, the seeds boost the cooperative process. For each sensor, the second stage lasts for at most two awake-sleep cycles. The sensors of type $d - 1 \leq x \leq k - 1$ enter in the second stage as soon as their 2-nd awake-sleep cycle starts. The sensors of type $0 \leq x \leq d - 2$ skip their second cycle and enter in the second stage within their 3-rd awake-sleep cycle. During the second stage, the seeds broadcast their corona identity for two awake periods if they have type $d - 1 \leq x \leq \lfloor 2d - 3 \rfloor_k$, or for one awake period, otherwise. The awake untrained sensors are listening until they become either *trained* or *white-flag*.

An untrained sensor that receives all concordant messages from its neighbors becomes trained and broadcasts for the remaining time slots of its awake period. Contrary, if an untrained sensor hears noise, that is, it receives more than one message from two or more neighbors transmitting different corona identities, it becomes a white-flag. It stops to listen and it waits the third stage to eventually acquire an approximation of its location.

The third stage of the sensor training protocol is also distributed and lasts for a single awake-sleep period for each sensor. Each trained sensor, which belongs to an even corona, transmits its corona identity, whereas all the awake white-flag sensors are listening. Since a white-flag is a sensor that in the second stage has received simultaneously two consecutive corona identities, it is surely covered by a sensor in an even corona. Such a sensor trains the white-flag during the 3-rd stage. Hence, at the end of the third phase, all the white-flag sensors are trained and they learn to belong to an even corona. Thus, the white-flags that belong to an odd corona acquire a localization that differs of at most ± 1 from the actual one. As a macroscopic effect, at the end of the third stage, the even coronas of the virtual infrastructure will expand over the odd coronas. It is worth noting that this approximation has little effects on the estimate of the distance from the sensors to the actor. Indeed, recalling that the sensors uses a transmission radius $r < 1/2$ during the training protocol and a transmission radius $r = 1$ to route messages from the sensors to the actor, a wrong sensor, which believes to be in corona γ but it is indeed in corona $\gamma \pm 1$, is at most at one extra hop from the actor.

Note that an untrained sensor that at the end of the second stage has heard nothing will not be involved in the third stage and it will remain untrained. Finally, sensors that acquire a localization that differs of more than ± 1 from the actual one are called *mistrained*.

However, as experimentally tested, if the network is sufficiently dense, very few sensors become mistrained or remain untrained.

4 Algorithm Properties

In order to analyze which sensors become seeds in the first stage in each corona, let us recall that, the sensors of type x , with $x \in [0, \dots, k-1]$, start the first awake period at the global time slot x and stay awake up to time $x+d-1$, while the actor broadcasts beacons $|k-1-x|_k, |k-1-x-1|_k, \dots, |k-1-x-d+1|_k$. Note that the sensors of type x receive the same beacons independent of the corona to which they belong, but they behave differently from one corona to another. In fact:

Lemma 1. *The seed in corona γ , $1 \leq \gamma \leq k-1$, are the sensors of type $x = |k-1-\gamma-w|_k$ with $w = [0, d-2]$, or equivalently:*

$$x \in \begin{cases} [|k-\gamma-d+1|_k, |k-1-\gamma|_k] & \text{if } |k-\gamma-d+1|_k \leq |k-1-\gamma|_k \\ [|k-\gamma-d+1|_k, k-1] \cup [0, |k-1-\gamma|_k] & \text{if } |k-\gamma-d+1|_k > |k-1-\gamma|_k \end{cases} \quad (1)$$

Similarly, the seeds in corona 0 are those with type $x = |k-1-w|_k$ with $w = [0, d-1]$, or:

$$x \in [|k-d|_k, |k-1|_k] \quad (2)$$

□

The second stage lasts $2k$ time slots, starting from the global time slot $k+d-1$. Recalling that a sensor of type x wakes up for the i -th awake period, with $i \geq 1$, at time slot $x+(i-1)L$, and that $L = k$, in the interval $t \in [k+d-1, 2k+d-2]$, all types of sensors enter in the second stage. In fact, at time t , the sensors of type $x = |t|_L = |t|_k$ wake up. Thus, during the interval $t \in [k+d-1, 2k-1]$ the sensors of type $x \in [d-1, k-1]$ wake up because they enter in the second stage in their 2-nd awake period, while during the period $t \in [2k, 2k+d-2]$ those of type $x \in [0, d-2]$ wake up because they enter in the second stage during their 3-rd awake period. Moreover:

Lemma 2. *In the interval $t \in [k+2d-2, 2k+2d-3]$, all the sensors of the $d-1$ types $|t-w|_k$, with $w = [0, d-2]$, are awake simultaneously.* □

While so far the results were independent of the network density, in what follows, the density plays an important role.

The cooperative process becomes operative in each corona when all the seeds are awake and broadcast. Thus, this happens for the first time, by Lemma 1, in corona $\gamma = |k-2d+2|_k$ at time slot $k+2d-3$. Since by the training condition TC2 all the seeds are awake simultaneously for two time slots, the seeds in corona $\gamma = |k-2d+2|_k$ are awake simultaneously and broadcast also at time slot $k+2d-2$. At that time, the sensors of the type $|2d-2|_k$, which are untrained in corona $|k-2d+2|_k$, wake up and, listening to their seed neighbors, they become trained. Then, the new trained sensors start to broadcast for the remaining $d-1$ time slots of their awake period, replacing the seed of type $|t-d+1|_k = d-1$ that go back to sleep. This is repeated for $k-d$ time slots up to time $2k+2d-3$, training all the type of sensors in corona $|k-2d+2|_k$.

Hence, during the cooperative process, an untrained sensor becomes trained only if it has in its neighborhood at least one trained sensor awake at the same time. This might happen or not depending on the network density. From now on, we assume that the network is sufficiently dense for the above condition to be verified. For the same reason, we consider the cooperative process to be operative starting from the time slot when all the $d - 1$ seeds are simultaneously awake. We discuss the effects of density only in Section 5, where experiments with different densities are reported.

Theorem 1. *Assuming that the network is sufficiently dense, the cooperative training process becomes effective in corona $\gamma = |k - 2d + 2 - y|_k$ at time slot $k + 2d - 2 + y$ and, in such a corona, a new type of sensors is trained in each subsequent time slot $2k + d - 2 + y$, with $0 \leq y \leq k - 1$. \square*

Observe that the last corona to be trained is corona $|k - 2d + 3|_k$ where the process lasts from time $2k + 2d - 3$ up to $3k + d - 3$. Moreover, note that at time slot $3k + d - 3$ the sensors of type $d - 2$, which entered as last in the second stage, have just completed their third sleep-awake cycle.

So far, it has been assumed that during the second stage each untrained sensor receives concordant and correct corona identities. Nonetheless, since all the sensors of the same type are always awake simultaneously independent of the corona to which they belong, but their status (i.e., seed, untrained, trained) depend on their corona, it may happen that the sensors in the corona borders listen to sensors of the same type but in different status. Consider, for example, the sensors of type $\bar{x} = |k - \gamma - d + 1|_k$, with $0 \leq \gamma \leq k - 1$ during the second stage. When such sensors wake up, they start to broadcast in corona γ where they are seed, whereas they listen in corona $\gamma - 1$ where they are untrained. A sensor of type \bar{x} on the border of corona $\gamma - 1$ can receive only the corona identity γ and thus it acquires a wrong localization. In this case, however, the correct localization may still be derived exploiting the fact that the sensor in corona $\gamma - 1$ has received beacon $\gamma - 1$ in the first stage.

Unfortunately, this is not always the case that the right corona information can be retrieved. There are cases when the sensors cannot acquire the exact localization or no localization can be derived during the second stage of the training protocol. For example, consider an untrained sensor in the corona border that hears two trained sensors of the same type belonging to two different coronas. Such a sensor can only hear noise and it cannot be localized. It will be a white-flag sensor, and it has to wait the third stage to be trained.

Theorem 2. *At the end of the third stage, all the white-flag sensors in the even coronas are turned into trained sensors, while those on the odd coronas become ± 1 -trained. \square*

Finally, in order to evaluate the power consumption per sensor during the cooperative algorithm, observe that, when a sensor is awake, its micro-controller is active and its radio is listening, receiving, or transmitting. Contrary, when a sensor is sleeping, its micro-controller is not active, its timer is on, and its

radio is off. Let p_{awake} , p_{TX} , and p_{sleep} be the power consumption by a sensor when it is listening/receiving, transmitting, or sleeping, respectively. Since the radio startup and shutdown require a not negligible overhead, let p_{trans} denote the power consumption for a sleep/wake transition followed by a wake/sleep transition.

Observing that a sensor wakes up at most for 4 times, that it transmits at most for 3 awake periods if it is a seed, and that the entire algorithm lasts 4 awake periods for each sensor, the maximum power consumed p_{max} per sensor can be upper bounded as:

$$p_{\text{max}} < 4p_{\text{trans}} + dp_{\text{awake}} + 3dp_{\text{TX}} + 4(k - d)p_{\text{sleep}} \tag{3}$$

In conclusion, recalling that $2 \leq d \leq L = k$, one has:

Theorem 3. *The cooperative training process terminates in $O(k)$ time slots. During the training, each sensor is awake for $O(d)$ time slots and consumes at most $4p_{\text{trans}} + dp_{\text{awake}} + 3dp_{\text{TX}} + 4(k - d)p_{\text{sleep}}$ power. \square*

Comparing the above results with the literature, the new cooperative training process is as fast as the best synchronized training algorithm and it is the fastest asynchronous training algorithm [24]. Moreover, in this protocol, since d is independent of k , each sensor can stay awake for a very short interval of time, almost constant. Instead, each sensor is awake for $O(\log k)$ and up to $O(k)$ time slots in the synchronous and asynchronous protocols, respectively. However, one cannot forget that this new protocol is probabilistic (i.e., we are not sure that all the sensors will be trained), while the previous algorithms are deterministic (i.e., all the sensors are trained).

5 Experimental Tests

In this section, the performances of the cooperative training algorithm, shortly denoted with *Coop*, are experimentally evaluated when the network density varies with respect to the accuracy of the localization, and the power consumption per sensor. In the simulation, each corona has a unit width and N sensors are uniformly distributed within a circle of radius k , centered at the actor. Moreover, each sensor generates its type x , as an integer uniformly distributed in the range $[0, k - 1]$.

By varying the total number of sensors N , the number of coronas k , and the sensor radius r , we consider three different settings for our simulations. For each setting, let $\mathbb{E}(N_x) = \mathcal{O}(\frac{N}{k})$ be the expected number of sensors of the same type $x \in [0, k - 1]$ and $\delta = \mathcal{O}(\frac{N}{\pi k^2})$ be the network density, that is the expected number of sensors that belong to a unit area of the actor zone.

Moreover, we consider the constant $q = \frac{\frac{N}{k} r^2}{\log(\frac{N}{k}) k^2}$ which is approximately the ratio between the number of the sensors of the same type x in a circle of radius r and the logarithm of the overall number of the sensors of the same type $\log(\frac{N}{k})$. Roughly speaking, q is a measure of the connectivity of the network. Very

Table 1. Experiment settings

	N	k	r	$\mathbb{E}(N_x)$	δ	q
S_1	310000	8	$\frac{1}{5}$	38750	1541.8	2.29
S_2	700000	12	$\frac{1}{7}$	58333	1547.3	0.75
S_3	819200	32	$\frac{1}{4}$	25600	254.65	0.15

Table 2. Estimate of sensor power consumption

Sensor Mode	Power consumption
μC sleep with timer on	60 μW
μC switch on, radio startup	30 mW
μC switch off, radio shutdown	30 mW
μC active, radio idle listening	60 mW
μC active, radio TX	80 mW

informally, if $q > 1$, it means that in a circle of radius r there are $\log(\frac{N}{k})$ sensors of the same type, and hence the sensors of such a type have a minimum degree of about $\log(\frac{N}{k})$ and therefore the network of such nodes is $\log(\frac{N}{k})$ -connected [8].

Table 1 reports, the parameters N , k , r , as well as $\mathbb{E}(N_x)$, δ , and q for the three settings S_1 , S_2 and S_3 used in the experiments.

In the settings S_1 , S_2 and S_3 , assuming the corona width $\rho = 100$ meters, there are 0.15, 0.15, and 0.025 sensors per square meter, respectively. At the present state of the technology, small sensors, supporting communications in a range varying from 10 to 100 meters, like TinyNode 584 produced by Shockfish S.A. or T-node developed by SOWNet Technologies can be used for built such massive networks [5,9].

Moreover, in order to evaluate the power consumption per sensor during the Coop algorithm, Table 2 reports the power consumption, measured in the field, of a T-node in different operational modes [9] to have a realistic setting. The data refer to the power consumed operating using 10 dBm transmission power, and hence attaining a transmission range around twenty meters, at a low bandwidth of 75 Kbit/s. Note that such a bandwidth is sufficient because the sensors have to transmit just their corona identity, which consists of $O(\log k)$ bits.

The Coop algorithm has been tested on each setting fixing $L = k$ and varying the sensor awake period d between 2 and 10. In fact, as proved by Lemma 1 and Theorem 1, the awake period d influences the number of seeds in the first stage as well as the number of trained sensors that are awake and broadcast in each time slot of the second stage.

In order to evaluate the quality of the localization, observe that, at the end of the Coop algorithm, a sensor can be in one of the following *status*:

- **trained**, if it has learnt the actual corona to which it belongs
- **± 1 -trained**, if it has learnt to belong to a corona which differs of ± 1 from the correct one

- **mistrained**, if it has acquired a corona which arbitrarily differs from the correct one
- **white-flag**, if it cannot decide to which corona it belongs although it is in the neighborhood of trained sensors
- **untrained** if it does not belong to the neighborhood of any trained sensor

In our experiments, statistics are taken on how many sensors are in each status at the end of the Coop algorithm. Specifically, Figures 1 and 2 report the results of the experiments on settings S_1 and S_2 along with S_3 , respectively, when the awake period d varies, with $d \geq 2$. The results are averaged over 3 independent experiments, which differ in the deployment distribution of the sensors and in the sensor type generation.

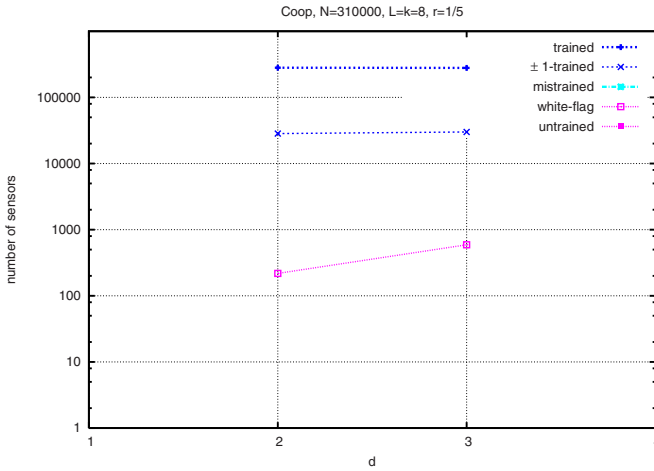


Fig. 1. Sensor statistics at the end of the Coop algorithm on setting S_1

At first, it is worth noting that there are no untrained sensors for all the experiments on the settings S_1 and for the experiments on S_2 with $d \geq 3$. Moreover, on S_3 , the number of untrained sensors rapidly decreases when d increases. This confirms the first setting has enough sensors to satisfy the density assumed in Theorem 1. Settings S_2 and S_3 are not sufficiently dense when $d = 2$. Increasing d , however, the number of untrained sensors decreases up to 0. We can say that the density assumed in Theorem 1 is achieved for S_2 and S_3 when $d = 3$ and $d = 9$, respectively. Not surprisingly, since S_3 has a value of q smaller than the one of S_2 , a greater value of d is needed. Indeed, in all the experiments when $(d - 1)q > 1$, at the end of the Coop algorithm, more than 98% of the sensors acquire a satisfying localization (i.e. trained or ± 1 -trained), and the 89% are correctly trained. In Table 3, for a detailed analysis of the trained sensors, the number of seeds, sensor trained, ± 1 -trained, and white-flag sensors are reported

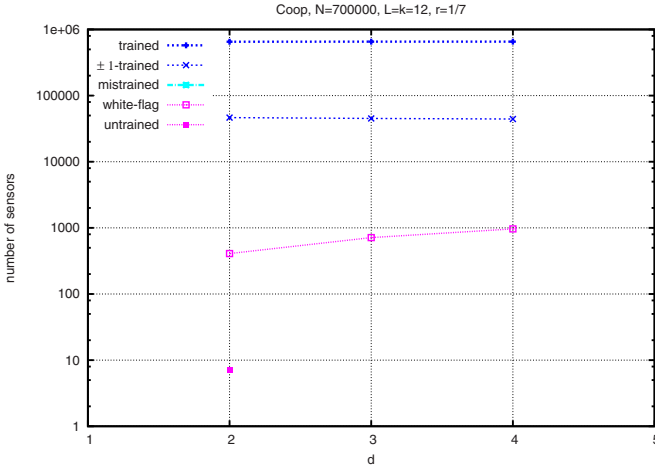


Fig. 2. Sensor statistics at the end of the Coop algorithm on setting S_2

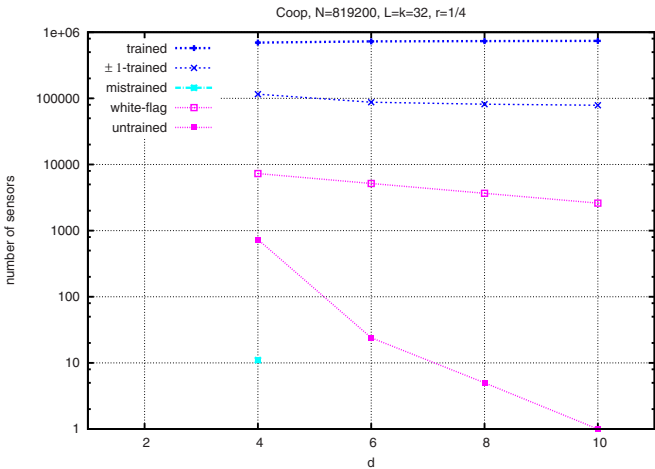


Fig. 3. Sensor statistics at the end of the Coop algorithm on setting S_3

at the end of the second stage (Coop2) of the training algorithm as well as at the end of the third stage (Coop). Moreover, half of the sensors which are white-flag at the end of the second stage become trained at the end of the third stage and half become ± 1 -trained.

The white-flag and ± 1 -trained sensors are placed at the borders of the coronas. When d increases, since more sensors are awake simultaneously the number of white-flag sensors increases, while that of the ± 1 -trained sensors decreases.

Table 3. The acquired localization in S_2 and S_3 after Coop2 and Coop

		$S_2, d = 2$	$S_2, d = 3$	$S_2, d = 4$	$S_3, d = 4$	$S_3, d = 6$	$S_3, d = 8$	$S_3, d = 10$
Coop2	trained	624787	620148	618647	639778	662189	666610	669426
	seed	58731	117131	175377	76934	127828	179162	230158
	± 1 -trained	15610	8176	5677	62218	24771	15213	10337
	white-flag	59596	71676	75676	116470	132216	137372	139436
Coop	trained	653052	654238	654735	695667	726559	733865	737996
	seed	58731	117131	175377	76934	127828	179162	230158
	± 1 -trained	46532	45050	44302	115506	87460	81664	78598
	white-flag	409	712	963	7293	5157	3664	2605

Table 4. Acquired localization in A-Seed and Coop2

	$S_1, d = 2$		$S_2, d = 2$	
	A-Seed	Coop2	A-Seed	Coop2
trained	260535	263877	623201	624787
± 1 -trained	823	6657	5751	15610
white-flag	48642	39466	71036	59596
untrained	0	0	12	7

It is worthy to note that we do not report the mistrained sensors because they are zero in all experiments but one. Thus, as expected, the network density guarantees that the corona identity propagation is confined in each corona.

About the ± 1 -trained sensors, their number also depends on the fact that the cooperative process does not start simultaneously in all the coronas because different coronas have different seeds. In fact, sensors at the border that wake up earlier than the seeds on their own corona might be ± 1 -trained. This behavior has been tested in Table 4, where a new algorithm, called *A-Seed*, is introduced. The A-Seed algorithm performs only the second stage of the cooperative training algorithm and assumes that the seeds are the sensors of a given type $\bar{x} \in [0, k-1]$, selected initially at random. Clearly, during the A-Seed algorithm, the cooperative process becomes effective at the same time slot in all the coronas and the number of white-flag sensors increases at the expenses of that of the ± 1 trained.

With respect to the power consumption, substituting the values in Table 2 in Equation 3, the maximum power consumed p_{\max} per sensor can be upper bounded as:

$$p_{\max} < 4 * 2 * 30 + d * 60 + 3 * d * 80 + 4 * (k - d) * 0.060mW \quad (4)$$

The results of the experiments on settings S_2 and S_3 are reported in Figures 4.

The behavior of the Coop algorithm is compared with that of the asynchronous training protocol Flat [2]. The Flat and Coop algorithms assume the same parameter values, except that Flat uses an awake-sleep cycle of length $L = k + 1$ instead of k . Indeed, when $L = k$, Flat cannot complete the training process [2], and thus the smallest value of L for which Flat trains all the sensors has been used.

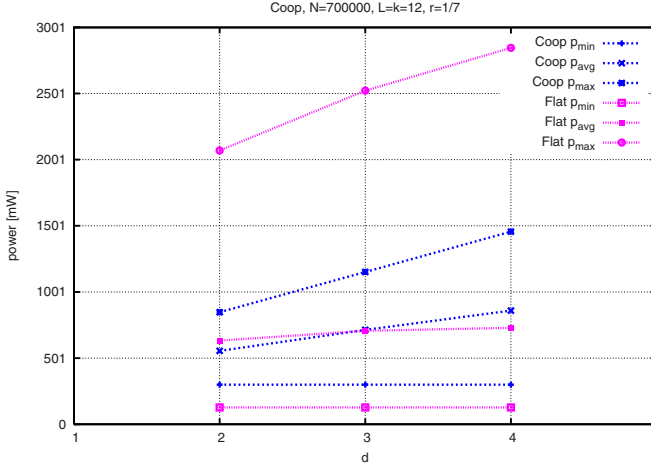


Fig. 4. Power consumption per sensor during the Coop algorithm on setting S_2

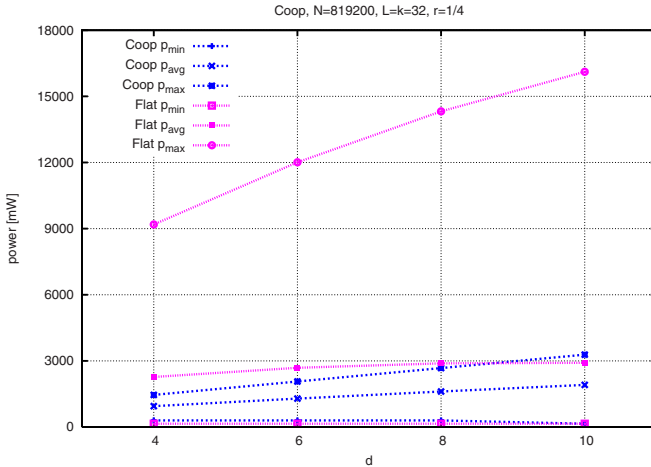


Fig. 5. Power consumption per sensor during the Coop algorithm on setting S_3

For each algorithm, it has been measured the maximum p_{max} (minimum p_{min} , resp.) power consumed by each sensor along with the average p_{avg} power.

One can note that although Coop and Flat consume overall almost the same power as shown in Figure 4, the difference between the sensor maximum and minimum power consumption in the Coop algorithm is much less than that measured for Flat. In other words, the Coop training algorithm drain the sensors in a balanced way, and therefore it works in favor of the network lifespan.

When k increases, like in Figure 4 scenario S_3 , the power effectiveness of the Coop algorithm is neat. The power p_{\max} of Coop is smaller than or equal to the p_{avg} of the Flat Algorithm for any value of d .

6 Conclusion

In the context of anonymous, asynchronous and randomly distributed sensor and actor networks, we have proposed a new cooperative training algorithm which exploits the high density features of the considered kind of network. After describing the phases of the algorithm, we have provided analytical and experimental results with respect to the accuracy for the localization and the consumed energy. The new training algorithm is particularly suitable in large and dynamic networks, that need to frequently and quick raise up the network, for instance in presence of an actor moving to track an intruder. Moreover, once the proposed course-grain localization has been performed, easy routing algorithms can be applied with respect to the obtained virtual infrastructure induced by our algorithm.

As an open problem, it remains to study analytically the minimum sensor network density which guarantees that the algorithm trains, with high probability, all the sensors. Moreover, as future works, one can investigate larger networks where the actors move. Even more challenging would be the comparisons of Flat and Coop in a test-bed wireless sensor network.

References

1. Akyildiz, I.F., Kasimoglu, I.: Wireless sensor and actor networks: research challenges. *Ad Hoc Networks* 2, 351–367 (2004)
2. Barsi, F., Bertossi, A.A., Betti Sorbelli, F., Ciotti, R., Olariu, S., Pinotti, M.C.: Asynchronous Corona Training Protocols in Wireless Sensor and Actor Networks. *IEEE Transactions on Parallel and Distributed Systems* (to appear)
3. Baryshnikov, Y.: Connectivity in Geometric Graphs: Beyond the Standard Model. *Private Communications*
4. Bertossi, A.A., Olariu, S., Pinotti, M.C.: Efficient corona training protocols for sensor networks. *Theoretical Computer Science* 402(1), 2–15 (2008)
5. Burri, N., von Rickenbach, P., Wattenhofer, R.: Dozer: Ultra-low power data gathering in sensor networks. In: *Proc. IPSN 2007*, Cambridge, MA (April 2007)
6. Gautschi, W.: The Incomplete Gamma Functions Since Tricomi. In *Tricomi's Ideas and Contemporary Applied Mathematics*. *Atti dei Convegni Lincei, Accademia Nazionale dei Lincei, Roma* 147, 203–237 (1998)
7. Olariu, S., Waada, A., Wilson, L., Eltoweissy, M.: Wireless sensor networks leveraging the virtual infrastructure. *IEEE Network* 18(4), 51–56 (2004)
8. Penrose, M.D.: *Random Geometric Graphs*. *Oxford Studies in Probability* (2003)
9. The Sensor Network Museum Project, <http://www.snm.ethz.ch/Main/HomePage>
10. Temme, N.: Uniform asymptotic expansions of the incomplete gamma functions and the incomplete beta functions. *Math. Comput.* 29, 1109–1114 (1975)
11. Temme, N.: The asymptotic expansion of the incomplete gamma function. *SIAM J. Math. Anal.* 10, 757–766 (1979)

12. Waada, A., Olariu, S., Wilson, L., Eltoweissy, M., Jones, K.: Training a wireless sensor network. *Mobile Networks and Applications* 10(1), 151–168 (2005)
13. Xu, Q., Ishak, R., Olariu, S., Salleh, S.: On asynchronous training in sensor networks. In: Lumpur, K. (ed.) *Proc. 3rd Intl. Conf. on Advances in Mobile Multimedia* (September 2005)
14. Xue, F., Kumar, P.R.: The number of neighbors needed for connectivity of wireless networks. *Wireless Networks* 10, 169–181 (2004)

Appendix

Procedure *Sensor*

Input: x, d, L, k, r, j ;

1. **case** j :
1. $j = 1$:
2. **for** $t := 0$ **to** $d - 1$ { *Initialize* }
3. $C[t] := -1$;
4. $\text{trained} := \text{white-flag} := \text{seed} := \text{false}$;
5. $\text{corona} := -\infty$; $\tau := -1$;
6. **for** $t := 0$ **to** $d - 1$ { *First stage* }
7. $C[t] := \text{listen-actor}(\gamma)$;
8. **if** $C[t] = 0$
9. **then** $\text{trained} := \text{seed} := \text{true}$; $\text{corona} := 0$;
10. **else if** $(t \geq 1 \text{ and } C[t] = \emptyset \text{ and } C[t - 1] = \gamma)$
11. **then** $\text{trained} := \text{seed} := \text{true}$, $\text{corona} := C[t - 1]$;
12. $\tau := \tau + 1$;
13. **if** $x \geq d - 2$
14. **then** $j := 2$; $\text{set-alarm-clock}(\tau + L - d)$;
15. **else** $j := 3$; $\text{set-alarm-clock}(\tau + 2L - d)$;
16. $j = 2, 3$:
17. **for** $i := j$ **to** 3 { *Second stage* }
18. **for** $t := 0$ **to** $d - 1$
19. **if** trained
20. **then** $\text{broadcast}(\text{corona})$
21. **else if** $\neg \text{white-flag}$
22. **then** $\text{listen-sensor}(\text{corona})$;
23. **if** $\text{corona} \neq \emptyset$ **then** $\text{corona} := \text{compatible}(\text{corona})$; $\text{trained} := \text{true}$;
24. **if** $\text{corona} = \text{noise}$ **then** $\text{white-flag} := \text{true}$;
25. $\tau := \tau + 1$;
26. **if** $(\text{white-flag} \text{ or } (\text{trained} \text{ and } \neg \text{seed}) \text{ or } (\text{seed} \text{ and } x \notin [d - 1, |2d - 3|_k]))$
27. **then** $j := 4$; $\text{set-alarm-clock}(\tau + (3 - i)L + L - d)$;
28. **else** $j := j + 1$; $\text{set-alarm-clock}(\tau + L - d)$;
29. $j = 4$:
30. **for** $t := 0$ **to** $d - 1$ { *Third stage* }
31. **if** trained **and** $|\text{corona}|_2 = 0$
32. **then** $\text{broadcast}(\text{corona})$
33. **else if** white-flag
34. **then** $\text{listen-sensor}(\text{corona})$;
35. **if** $\text{corona} \neq \emptyset$ **then** $\text{trained} := \text{true}$;
36. $\tau := \tau + 1$;
37. $\text{set-alarm-clock}(\tau + L - d)$;

Fig. 6. The corona training protocol for the sensor

Multi-Agent Itinerary Planning for Wireless Sensor Networks

Min Chen¹, Sergio Gonzalez², Yan Zhang³, and Victor C.M. Leung²

¹ School of Computer Science & Engineering, Seoul National University, 151-744, Korea
mchen@mmlab.snu.ac.kr

² Elect & Comp Eng, University of British Columbia, V6T 1Z4, Canada
sergiog,vleung@ece.ubc.ca

³ Simula Research Laboratory, 1325 Lysaker, Norway
yanzhang@ieee.org

Abstract. Agent-based data collection and aggregation have been proved to be efficient in wireless sensor networks (WSNs). While most of existing work focus on designing various single agent based itinerary planning (SIP) algorithms by considering energy-efficiency and/or aggregation efficiency, this paper identifies the drawbacks of this approach in large scale network, and proposes a solution through multi-agent based itinerary planning (MIP). A novel framework is presented to divide our MIP algorithm into four parts: visiting central location (VCL) selection algorithm, source-grouping algorithm, SIP algorithm and its iterative algorithm. Our simulation results have demonstrated that the proposed scheme lowers delay and improves the integrated energy-delay performance compared to the existing solutions with the similar computation complexity.

Keywords: Wireless sensor networks, mobile agent, itinerary planning.

1 Introduction

The application-specific nature of a wireless sensor network (WSN) requires that sensor nodes have various capabilities. It would be impractical to store all the programs needed in the local memory of embedded sensors to run every possible application, due to the tight memory constraints. The intrinsically flexible features of mobile agent (MA) make it adaptable to diverse network conditions in dynamically reconfigurable WSNs.

An agent deployed in a sensor network is a special kind of software that migrates among network nodes to carry out a task autonomously, in order to achieve the objectives of the sink node.

Compared to its traditional client/server computing communications mechanism counterpart, mobile agent based computing has exhibited its unique efficiency in context-aware sensory environments [1, 2, 3, 4, 5, 6, 7, 8]. In a previous survey [1], we separated the agent design process for WSNs into four parts: architecture, itinerary planning, middleware system design, and agent cooperation for the design, development, and deployment of MA systems for high-level inference and surveillance in WSNs.

Among the four components, itinerary planning determines the order of nodes to be visited during agent migration, which has a significant impact on energy performance

of the MA systems. Though the agent itinerary is critical to the network performance, it has been shown that finding an optimal itinerary is NP hard and still an open area of research. Therefore, heuristic algorithms [2, 4, 10] and genetic algorithms [5] are generally used to compute itineraries with a sub-optimal performance. Though our previously introduced IEMF and IEMA approaches [10] exhibit higher performance in terms of energy efficiency and delay compared to the existing solutions, the limitation of utilizing a single agent to perform the whole task, making the algorithm unscalable in applications with a large number of source nodes needed to be visited. Typically, single agent itinerary planning algorithms have high efficiency in the applications with the following characteristics:

- The source nodes are distributed geographically close to each other.
- The number of source nodes is not large.

For a large scale sensor networks, with many nodes to be visited, single agent data dissemination exhibits the following pitfalls:

1. *Large Delay*: Extensive delay is needed when a single agent works for networks comprising hundreds of sensor nodes.
2. *Unbalanced load*: There are two kinds of unbalancing problems while using a single agent. First, in the perspective of the whole network, all of the traffic load is put on a single flow. Therefore, sensor nodes in the agent itinerary will deplete energy quickly than other nodes. Secondly, from the perspective of the itinerary, the agent size increases continuously while it visits source nodes, and so the agent transmissions will consume more energy in its itinerary back to the sink node.
3. *Insecurity with large accumulated size*: The increasing amount of data accumulated by the agent during its migration task increases its chances of being lost due to noise in the wireless medium. Thus, the longer the itinerary, the higher risky of the agent-based migration becomes.

In this paper, we propose a novel Multi-agent Itinerary Planning (MIP) algorithm to address the above issue. Traditionally, Single-agent Itinerary Planning (SIP) includes the following two challenges:

- Selecting the set of the source nodes to be visited by the mobile agent.
- Determining a node visiting sequence in an energy-efficient manner.

Compared to existing SIP proposals, the main contributions of this paper are listed as follows:

- We introduce a novel source-grouping algorithm. Note in [11], clustering based architecture is utilized to facilitate mobile agent based data dissemination. Though our source-grouping algorithm partitions source nodes into several sets, which has a similar effect of grouping source nodes in clusters, we do not set up a hierarchical structure. Thus, our algorithm does not have any control message overhead for the clustering process.
- We propose an iterative algorithm for MIP solution.
- We propose a generic framework to design a MIP algorithm. Within this framework, any SIP algorithm can be extended to the corresponding MIP algorithm, where the SIP algorithm will be carried out iteratively until the source list is empty.

The remainder of the paper is organized as follows. The problem is stated in Section 2. We present the proposed MIP algorithm in Section 3. Our simulation studies are reported in Section 4. Section 5 concludes the paper.

2 Problem Statement

2.1 Motivation

In this section, the motivation for MIP proposal is illustrated through Eqns. (1) and (2). The agent size at the k th source depends on three parts: (1) the initial agent size (l_{ma}^0), which includes size of processing code and agent header; (2) size of reduced payload when visiting the first source node ($l_{data} \cdot (1 - r_1)$), where r_1 is the data reduction ratio at the first source. Note that there is no data aggregation at the first source; (3) accumulated size of the aggregated data payload after local processing from the second source node to the present source ($\sum_{i=2}^k l_{data} \cdot (1 - r_i) \cdot (1 - \rho_i)$). Thus, the final agent size increases linearly with the source number, as shown in Eqn. (1).

$$l_{ma}^k = l_{ma}^0 + l_{data} \cdot (1 - r_1) + \sum_{i=2}^k l_{data} \cdot (1 - r_i) \cdot (1 - \rho_i). \tag{1}$$

$$E_{itinerary} = E_0^1 + \sum_{k=2}^n E_{k-1}^k(l_{ma}^{k-1}) + E_n^0. \tag{2}$$

Table 1. Notation

Symbol	Definition
l_{data}	the size of raw sensory data at a source node.
l_{ma}^0	the size of mobile agent when dispatched from the sink.
r_i	the reduction ratio at the k th source by agent assisted local processing.
ρ_i	aggregation ratio at the k th source by agent for data redundancy elimination.
l_{ma}^k	the agent size when it leaves the k th source.
N	the number of source nodes needed to be visited.
$E_{k-1}^k(l_{ma})$	the communication energy cost during a mobile agent roams from source $k-1$ to source k with agent size l_{ma} .

Fig. 1 presents a typical scenario of single agent based data dissemination. In order to calculate the itinerary cost ($E_{itinerary}$), we divide the whole itinerary cost into three parts: (1) from the sink node to the first source node S_1 , only the processing code and

¹ Please refer Table 1 for the definitions of ρ_i and l_{data} .

agent header are included in the MA packet. We denote the communication energy consumption in this part by E_0^1 ; (2) the second part starting from the time when MA leaves the first source node to the time when it visits the last source node S_n . The communication energy consumption in this phase is denoted by $\sum_{k=2}^n E_{k-1}^k(l_{ma}^{k-1})$, where $E_{k-1}^k(l_{ma}^{k-1})$ represent the communication energy cost for the MA to roam from source $k-1$ to source k with agent size l_{ma} ; (3) the third part starting from the time when MA finishes visiting all the source nodes to the time when it returns to the sink. The communication energy consumption in this part is denoted by E_n^0 . Eqn. (2) shows that the itinerary cost is a squarely increasing function of the source node number, which causes the performance of the SIP algorithm to deteriorate in large scale sensor networks. The end-to-end agent delay exhibits a trend that is congruent to the similar trend as the itinerary cost. Thus, we are motivated to design a MIP algorithm that possesses the flexibility of adapting to the specific network parameters, such as network size, source node number, reduction ratio, aggregation ratio, sensor data size, etc. Specifically, a SIP algorithm can be deemed as a particular output of the MIP algorithm with a single agent.

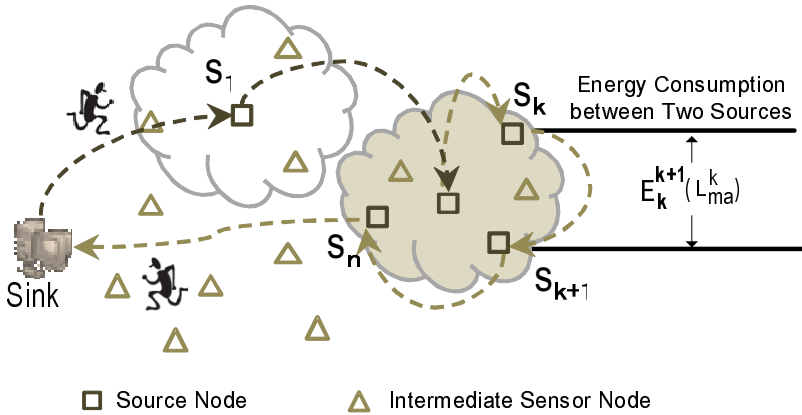


Fig. 1. Illustration of Single Agent based Itinerary Planning

2.2 A Generic Multi-Agent Itinerary Planning Algorithm

We state our assumptions and define a generic MIP algorithm in this section as follows:

- primary itinerary design algorithms are executed at the sink, which has relatively plenty of resources in terms of energy and computation²
- the sink node knows the geographic information of all the source nodes. Note that in our algorithm, only source locations are needed, while the other algorithms [4, 5] need all of the nodes' geographical positions.

In fact, the above assumptions are common in most of the solutions presented in [4, 5, 10] for the SIP problem. The previous SIP algorithms assume that the set of source

² MAs may deal with unexpected failures of arriving next source nodes, as soon as failure is detected, MA change the source destination node scheduled to be visited after the failed node.

nodes to visit is predetermined. In contrast, our MIP algorithm needs to group source nodes for different mobile agents, since the determination of source visiting set is a dynamic process. The proposed MIP algorithm can be deemed as the iterative version of a SIP solution, which can be divided into four parts:

- *Selection of Visiting Central Location (VCL) for an agent:* While using multiple agents, it is a challenging issue to use the least number of them while achieving the required coverage of source nodes. Strategically, the agent's VCL is selected to the center of area with a high source node density. Finding an optimal agent number is also a NP-hard problem.
- *Determining the source visiting set:* In order to determining the source visiting set, we first isolate the visiting area, which is typically a circle/oval centered at the VCL and it has a certain radius. All of the source nodes in the disk will be included in the visiting list of the agent.
- *Determining a source-visiting sequence:* This is the itinerary plan for the current agent. In this step, the problem is simplified into the *Single-agent Itinerary Planning* problem, whereby existing SIP solutions can be applied, such as LCF, GCF, MADD, IEMF and IEMA, etc.
- *Algorithm iteration:* If there are source uncovered source nodes, the next VCL will be calculated based on the remaining set of source nodes. The previous process will repeat until all of the source nodes have been assigned to a mobile agent.

3 Proposed MIP Algorithm

VCL-Selection Algorithm. The basic idea of the proposed *visiting central location* (VCL) selection algorithm is to distribute each source's impact factor to other source nodes. Let n denote the source number. Then, each source will receive $n - 1$ impact factors from other source nodes, and one from itself. After calculating the accumulated impact factor, the location of the source with the largest accumulated impact factor will be selected as VCL.

We achieve this by using the analogy of a gravity field: a source node is modeled by a small iron ball, and the network is seen as an elastic plane, as shown in Fig. 2(a). When the iron ball is put on the elastic plane, the plane will be naturally distorted to the shape as shown in Fig. 2(b). In our approach, we map the physical model to a sensor network in a way where each source will contribute with a certain gravity impact to a fixed location. If we overlap all of the source nodes' gravity fields, there must be a location suffering the largest gravity³. We define this location as VCL.

However, there are unlimited locations in the plane. Therefore, in order to reduce computational complexity, we make the following simplifications:

- the gravity impact is quantized by hop count between two source nodes.
- only the location of a source node is considered as a candidate to be selected as VCL.

³ In Physics, this is analogous to a Boltzmann Machine, and gradient descent.

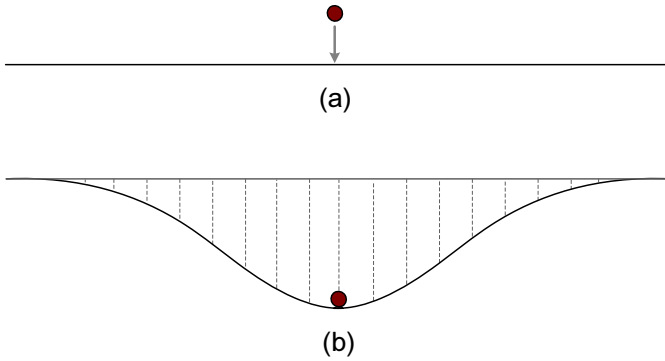


Fig. 2. Analog to Illustration of Calculating the Impact Factor between Two Source Nodes

We denote the set of n nodes by V_n . For any two source nodes $i, j \in V_n$, d_{ij} denotes the distance between i and j . Then, we can estimate the hop count between i and j as $H_{ij}^j = \lceil \frac{d(k-1, k)}{R} \rceil$, where R represents the maximum transmission range. To approximate the effect of a real gravity field, a gauss function is adopted to calculate the impact factor between i and j :

$$G_{ij} = e^{-\frac{(H_{ij}^j - 1)^2}{2\sigma^2}} \tag{3}$$

Fig. 3 shows an example with σ set to 8. A suitable setting should be heuristically selected for different network scale and different requirements of grouping effect.

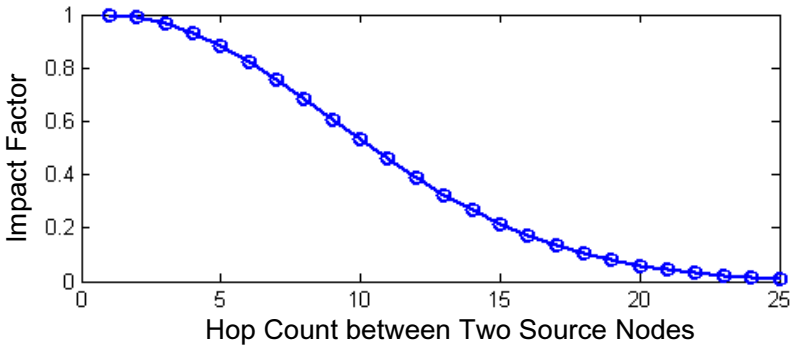


Fig. 3. Illustration of Calculating the Impact Factor between Two Source Nodes

The pseudo code of the VCL-selection algorithm is listed at Algorithm 1.

Source-grouping algorithm. Our source-grouping algorithm is very simple. Let $A(VCL, R)$ denote the circular area centered at VCL with a radius of R . Then, all of the source nodes within $A(VCL, R)$ will be included in the visiting list which is assigned to the current agent. Algorithm 2 shows the pseudo-code of the proposed source-grouping algorithm.

Algorithm 1. VCL-selection algorithm for the set of source nodes (V_m)

```

for each source  $i$  in  $V_m$  do
   $G_i \leftarrow 0$ ;
end for
for each source  $i$  in  $V_m$  do
  for each source  $j$  in  $V_m$  do
    calculate  $G_{ij}$  according to Eqn.(3);
     $G_i \leftarrow G_i + G_{ij}$ ;
  end for
end for
for each source  $k$  in  $V_m$  do
  if  $G_k = \min\{G_i | i \in V_m\}$  then
    select the position of node  $k$  as VCL;
    break;
  end if
end for

```

Algorithm 2. Source-grouping algorithm for the set of source nodes (V_m)

```

for each source  $i$  in  $V_m$  do
  calculate the distance ( $d_{vcl,i}$ ) between VCL and node  $i$ ;
  if  $d_{vcl,i} < R$  then
     $V_{left} \leftarrow V_m - i$ ;
     $V_{group} \leftarrow V_{group} + i$ ;
  end if
end for

```

Iteration based MIP Algorithm. For each iteration, a new VCL will be calculated for the remaining source list. Then, a new list of source nodes will be assigned to a mobile agent. To this moment, the itinerary for the agent can be planned by any SIP algorithms. In this paper, some typical SIP algorithms are tested, such as LCF, GCF and IEMF. If the remaining source list is not empty, the above process will repeat until all of the source nodes have been assigned to a mobile agent. The pseudo code of the iteration based MIP algorithm is shown at Algorithm 3.

Algorithm 3. MIP algorithm for the whole set of source nodes (V_n)

```

 $V_{left} \leftarrow V_n$ ;
loop
  if  $V_{left}$  is not empty then
    calculate VCL  $V_{left}$  according to Algorithm 1;
    calculate  $V_{group}$  according to Algorithm 2;
    perform SIP algorithm for  $V_{group}$ ;
    updated  $V_{left}$  according to Algorithm 2;
  end if
end loop

```

The computational complexity of Algorithms 1, 2 and 3 is $O(n^2)$, and the one for our MIP scheme depends on the SIP algorithm. For example, if a SIP (e.g., LCF) has

a computational complexity of $O(n^2)$, then a LCF-based MIP algorithm will have the same computation complexity.

4 Performance Evaluation

4.1 Simulation Setting

We implement the proposed MIP algorithm as well as the three existing SIP algorithms (LCF, GCF and IEMF) using OPNET Modeler, and perform extensive simulations. We choose a network where nodes are uniformly deployed within a $1000\text{m} \times 500\text{m}$ field. To verify the scaling property of our algorithms, we select a large-scale network with 800 nodes. We assume that the sink node is located at the right side of the field and multiple source nodes are randomly distributed in the network.

The sensor application module consists of a constant-bit-rate source, which generates a sensor data report every 1 s (1024 bits each). As in [10], we use IEEE 802.11 DCF as the underlying MAC, and the radio transmission range is set to 60 m. The data rate of the wireless channel is 1 Mb/s. All messages are 64 bits in length. For consistency, we use the same energy consumption model as in [9]. The initial energy of each node is 5 Joules. The power consumptions for transmission, reception and idling are 0.66 W, 0.395 W, and 0.035 W, respectively. We count for all types of energy consumptions in the simulations, including transmission, reception, idling, overhearing, collisions and other unsuccessful transmissions, MAC layer headers, retransmissions, and RTS/CTS/ACKs.

We consider the following four performance metrics:

- *Task Duration*: in a SIP algorithm, it is the average delay from the time when a MA is dispatched by the sink to the time when the agent returns to the sink. In our MIP algorithm, since multiple agents work in parallel, there must be one agent which returns to the sink at last. Then, the task duration of our MIP algorithm is the delay of that agent.
- *Average Communication Energy*: the total communication energy consumption, including transmitting, receiving, retransmissions, overhearing and collision, over the total number of distinct reports received at the sink.
- *Hop Count*: in SIP, it is the average hop count of a mobile agent itinerary. In MIP, it is the accumulated hop counts of all the agents.
- *Integrated Performance*: For time-sensitive applications over energy constrained WSNs, we consider both delay and energy performances, and evaluate the integrated performance (denoted by η) in terms of task duration and average communication energy. The smaller the value of η is, the better the integrated performance will be.

$$\eta = \text{energy} \cdot \text{delay}. \quad (4)$$

In all the figures presented in this section, each data point is the average of 25 simulation, which runs with different random seeds. The mobile agent specific parameters are shown in Table 2.

Table 2. Simulation Parameters

Raw Data Reduction Ratio (τ)	0.8
Aggregation Ratio (ρ)	0.9
MA Accessing Delay (τ)	10 ms
Data Processing Rate (V_p)	50 Mbps
Size of Sensed (Raw) Data (l_{data})	Default: 2048 bits
Size of Processing Code (l_{proc})	1024 bits
The Number of Source Node (n)	Default: 40
Radius to the center point	Default: 255

4.2 Simulation Results

Multiple Itineraries Construction in the Proposed MIP Algorithm. In this section, we will show the snapshot of OPNET simulation for MIP algorithm, and a typical SIP algorithm (i.e., LCF). Fig. 4 shows the result of source-grouping and itinerary planning for the first mobile agent. The circled node is selected as the first VCL, since it has the highest accumulated impact factor of 14.87. The second VCL has the highest impact factor of 4.14, which is much smaller than that of the first one. It is because the number of candidate source nodes is smaller than that of the first round selection, as shown in Fig. 5. Note that the number of left source nodes will become smaller and smaller, and only three source nodes are visited by the third agent, as shown in Fig. 6. Since the three agents are dispatched by the sink node in parallel, they collect sensory data concurrently. Observed from above figures, the third agent will return to the sink first, then the second agent. Finally, the task duration is actually the delay of the first agent. In order to compare MIP and SIP algorithms, the planned itinerary by LCF is plotted in Fig. 7. Intuitively, the length of itinerary is longer than that of the first agent in

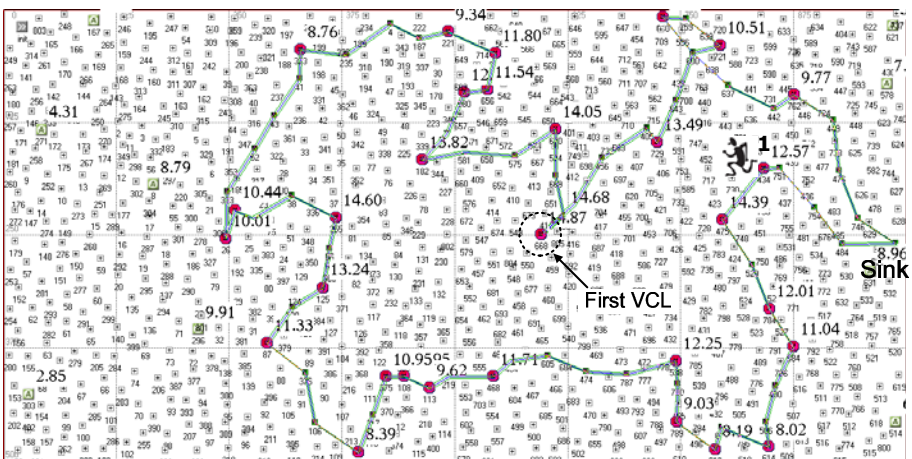


Fig. 4. The snapshot of the first itinerary in MIP algorithm

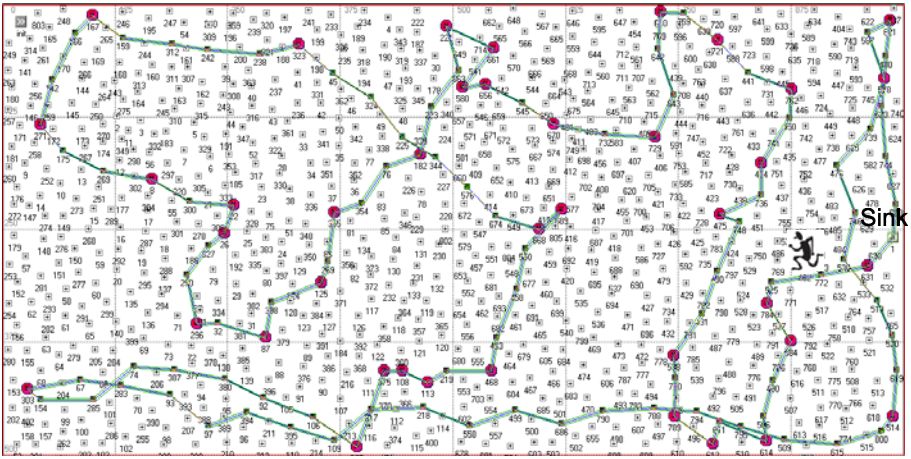


Fig. 7. The snapshot of LCF algorithm

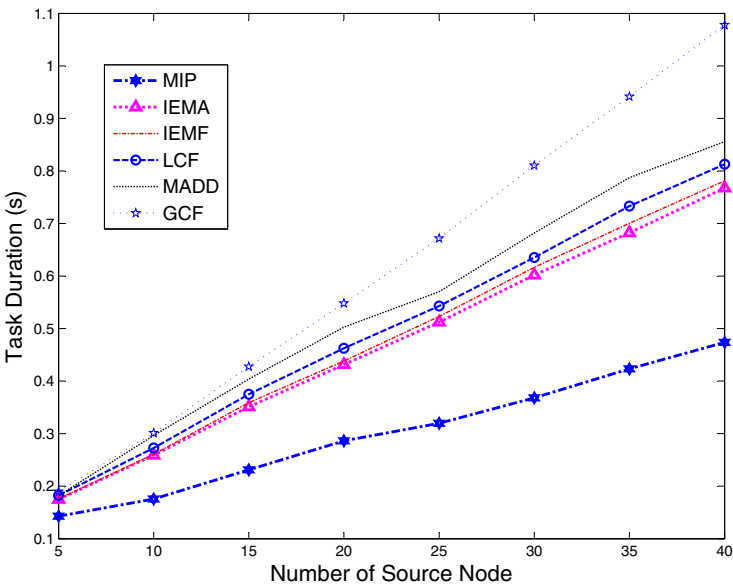


Fig. 8. Task durations

As shown in Fig. 8, MIP algorithm has absolute advantage in terms of task duration, which is only half of that of LCF. Note that the task duration of MIP is calculated fairly with SIP algorithms. Since we dispatch all of the mobile agents simultaneously, contention exists when multiple agents are close to each other. Even so, MIP still has superior delay performance than SIP algorithms.

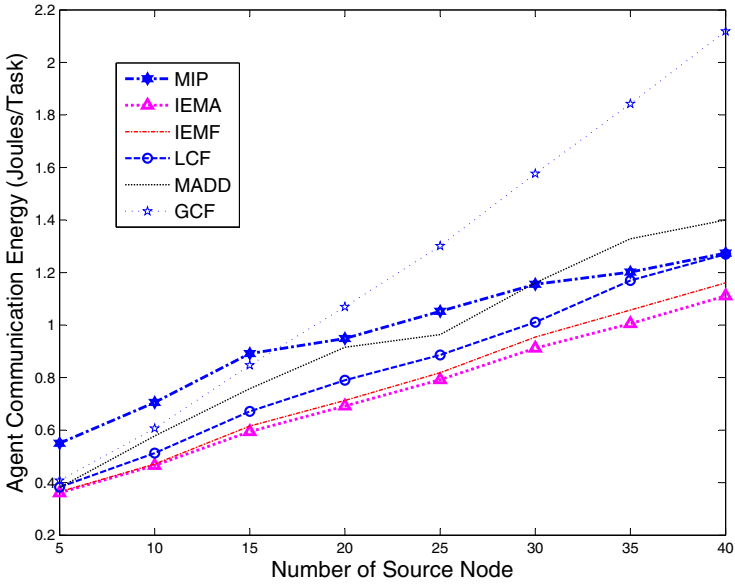


Fig. 9. Task communication energy

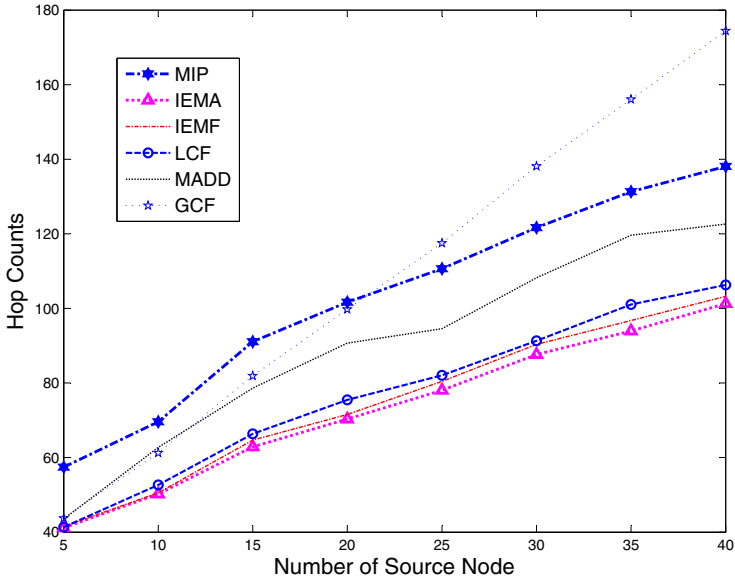


Fig. 10. Hop counts

In Fig. 9, the energy consumption of MIP algorithm is much higher than that of SIP algorithms when source number is small. Actually, it is only necessary for the usage of

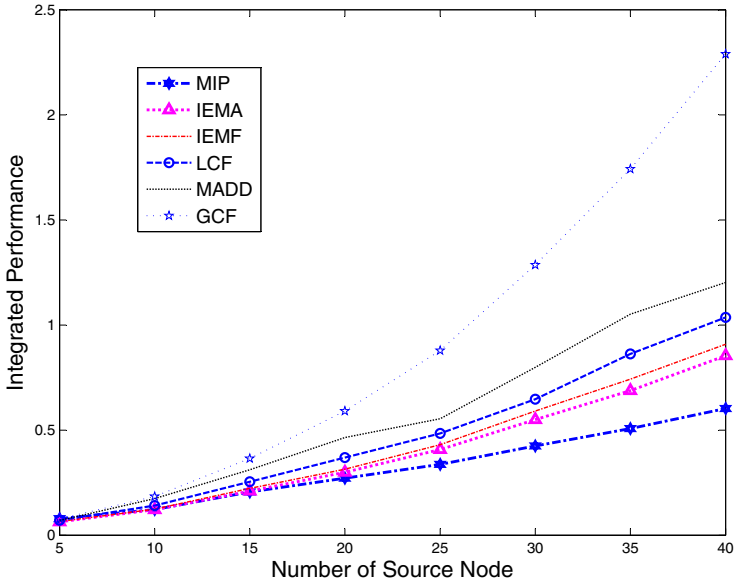


Fig. 11. Agent accumulated delay

multi-agent when source number is large. When the source number is 40, the energy consumption of MIP algorithm becomes comparable to those of SIP algorithms.

In Fig. 10, the accumulated hop counts of MIP algorithm is larger than the hop count of SIP. By comparison, it is important to consider the joint delay and energy performance, especially for delay constraint traffic in wireless sensor network, such as wireless multimedia sensor network, and video sensor networks [12]. Fig. 11 shows that MIP algorithm has the best integrated performance, which verifies effectiveness of the proposed algorithm.

5 Conclusions

In this paper, we addressed the problem of itinerary planning for multi-agent based data dissemination, facilitating concurrent sensory data collection to reduce task duration extensively. The proposed multi-agent itinerary planning (MIP) algorithm has the similar complexity with most of single agent based itinerary (SIP) algorithm, and can be flexibly adaptive to network dynamics in various network scales. We will propose more efficient source-grouping algorithm in our future work.

Acknowledgement. This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. This work was supported in part by the IT R&D program of MKE/IITA [2007-F-038-03, Fundamental Technologies for the Future Internet] and grant number IITA-2009-C1090-0902-0006. OPNET University Program and The ICT at Seoul National University provides research facilities for this study.

References

1. Chen, M., Gonzalez, S., Leung, V.C.M.: Applications and design issues of mobile agents in wireless sensor networks. *IEEE Wireless Communications* 14(6), 20–26 (2007)
2. Chen, M., Kwon, T., Yuan, Y., Choi, Y., Leung, V.: Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing* 2007(1), 219–242 (2007)
3. Chen, M., Kwon, T., Yuan, Y., Leung, V.C.M.: Mobile Agent Based Wireless Sensor Networks. *Journal of Computers* 1(1), 14–21 (2006)
4. Qi, H., Wang, F.: Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. In: *Proc. IEEE ICC 2001, Helsinki, Finland* (2001)
5. Wu, Q., Iyengar, S.S., Rao, N.S.V., Barhen, J., Vaishnavi, V.K., Qi, H., Chakrabarty, K.: On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Trans. Knowledge and Data Engineering* 16(6), 740–753 (2004)
6. Qi, H., Xu, Y., Wang, X.: Mobile-Agent-Based Collaborative Signal and Information Processing in Sensor Networks. *Proceedings of the IEEE* 91(8), 1172–1183 (2003)
7. Xu, Y., Qi, H.: Mobile agent migration modeling and design for target tracking in wireless sensor networks. *Ad Hoc Networks Journal* 6(1), 1–16 (2008)
8. Tseng, Y., Kuo, S., Lee, H., Huang, C.: Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *The Computer Journal* 47(4), 448–460 (2004)
9. Mao, S., Hou, Y.T.: BeamStar: An edge-based approach to routing in wireless sensor networks. *IEEE Trans. Mobile Computing* 6(11), 1284–1296 (2007)
10. Chen, M., Leung, V., Mao, S., Kwon, T., Li, M.: Energy-efficient Itinerary Planning for Mobile Agents in Wireless Sensor Networks. In: *IEEE International Conference on Communications (ICC 2009), Dresden, Germany* (2009)
11. Xu, Y., Qi, H.: Distributed Computing Paradigms for Collaborative Signal and Information Processing in Sensor Networks. *International Journal of Parallel and Distributed Computing* 64(8), 945–959 (2004)
12. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A Survey on Wireless Multimedia Sensor Networks. *Computer Networks* 51(4), 921–960 (2007)
13. http://en.wikipedia.org/wiki/Intrinsic_safety
14. <http://www.msha.gov/techsupp/PEDLocating/MSHAApprovedPEDproducts.pdf>
15. Liu, H., Lai, X., Ma, D.: Application of CDMA Technology based Communication in Underground Coal Mines. *Mobile Communication (China)* 29(10), 113–114 (2005)
16. Chen, M., Gonzalez, S., Zhang, Q., Li, M., Leung, V.: 2G-RFID based E-healthcare System. In: *IEEE Wireless Communications Magazine, Special Issue on Wireless Technologies for E-healthcare* (to appear)

Using Sensor Networks to Measure Intensity in Sporting Activities*

Mark Roantree¹, Michael Whelan², Jie Shi¹, and Niall Moyna²

¹ Interoperable Systems Group, Dublin City University
{mark,jshi}@computing.dcu.ie

² School of Health and Human Performance, Dublin City University
{niall.moyna,michael.whelan}@dcu.ie

Abstract. The deployment of sensor networks is both widespread and varied with more niche applications based on these networks. In the case study provided in this work, the network is provided by two football teams with sensors generating continuous heart rate values for the duration of the activity. In wireless networks such as these, the requirement is for complex methods of data management in order to deliver more and more powerful query results. In effect, what is required is a traditional database-style query interface where domain experts can continue to probe for the answers required in more specialised environments. This paper describes a system and series of experiments that requires powerful data management capabilities to meet the requirements of sports scientists.

Keywords: Wireless Sensor Network, Data Synchronisation, Calibration, Query Service.

1 Introduction

Sensor networks have become more varied with many niche applications based on a wide variety of application areas. Unobtrusive sensors are now commonly used to assess the physiological responses during individual and teams sports. The measurement of heart rate to assess the physiological load during individual and team sports is widely accepted within the sporting community [2,13]. Ambulatory telemetric equipment such as the wireless Polar Team Heart monitor [12] used in this study has made it possible to innocuously monitor heart rates during team sports. Relative exercise intensity can be estimated [1] by processing and manipulating the output from heart rate monitors as this is commonly used as a measure of exercise intensity during a game of soccer [2,13].

Gaelic football [14] is the most popular sport in Ireland. It is a hybrid of Rugby and Australian Rules football. This project assessed heart rate responses during small sided and regular Gaelic football games in young players. To achieve this, we created a wireless sensor network that has multiple configurations and

* Partially Funded by Enterprise Ireland Grant CFTD-2008-231.

requires a sophisticated data management layer to process, normalise and query the data streams.

This paper is structured as follows: in the remainder of this section we provide the motivation and contribution for our research; in §2 we describe the sensor network in terms of components and different configurations; in §3 we introduce a DataSpace architecture that provides the platform for data management in the sensor network; in §4 we describe an application to harvest data from sensor networks; in §5 a set of experiments with analysis of the results are given; in §6 we provide details of related research; while §7 offers conclusions.

1.1 Requirements and Motivation

To accurately determine the intensity at which each player is working during a game or training session, a calculation of each player's maximum heart rate (MHR) and resting heart rate (RHR) must be determined. Resting heart rate was determined following a 5 minute rest period and Maximal heart rate was determined using a field based test.

Once again, heart rate data can also indicate the amount of time spent in different intensity zones (table 1) but only with adequate data management techniques and a flexible query interface. Such a query interface will require user interaction with the sensor network and not a series of 'built-in' queries. Specifically, the development of a quick accurate measurement of the amount of time spent in each training zone is an important factor in determining the primary energy source utilised during games and training. In addition, it may also facilitate coaches in developing individual physiological profiles for each player. This will allow coaches to design and implement appropriate individual training programmes.

The requirement for an infrastructure to monitor and optimise players' performances led to the development of a wireless sensor network that was configured for each experiment as described in section §2. While the network provided the participants and hardware, it was then necessary to provide the data management layer in order to process and calibrate data generated by the networks. The motivation is to provide a traditional query interface for the low level data generated by the wireless network. The research described in this paper is a result of a collaboration between the Interoperable Systems Group (ISG) and the School of Health and Human Performance, both at Dublin City University in Ireland. While the usage of XML to provide interoperability for sensor networks

Table 1. Heart Rate Training Zones

Perc.	Zone	Description	Typical Range
Rest to 60%	Resting	Walking Pace	RHR to 120
60%-70%	Recovery	Develops basic endurance and aerobic capacity.	120 to 140
70%-80%	Aerobic	Develops the cardiovascular system.	140 to 160
80%-90%	Anaerobic	Develops the lactic acid system.	160 to 180
90%-100%	Maximal	Training in this zone is optimal for development of players' aerobic capacity but is possible only for short periods.	180 to 200

is gaining in popularity, XML has well known performance issues [6] [11] and the manipulation of this sensor network requires normalisation and continuous recalibration of the sensor data. Thus, any approach that uses XML as part of the data management level must demonstrate that queries times are within an acceptable level of performance.

1.2 Contribution

The construction of a sensor network comprising athletes and heart rate monitors, and configured in different ways provides a strong foundation for analysing the performance characteristics of athletes and in building personal and team-based physiological profiles. However, this represents the physical layer in the solution and a software layer comprising all of the data management and query performance aspects is still required. In this paper, we describe those data management components that facilitate user manipulation and analysis. Our classification and calibration services are the key enablers in the provision of a robust query service and these will be discussed in detail, providing algorithms for locating only meaningful data. Furthermore, we believe our usage of a DataSpace Architecture [5] facilitates the required heterogeneity in data management for sensors networks where data arrives in multiple formats, requiring multiple processors and different processing logic. Our prototype and experiments will demonstrate the speed at which data becomes available to users and the response times they can expect from queries.

2 Wireless Sensor Network Configurations

In this section, we describe the various configurations of the wireless sensor network and outline the underlying scenarios giving rise to these configurations. The actual experiments involved a number of school-aged football teams playing Irish Gaelic football where each player wore a heart monitor that monitored and broadcasted their heart-rate values, every 5 seconds, to a base station. The configuration of the sensor network was determined by the context for each experiment. These are the contexts in which players wore the heart monitors.

- 15-a-side games. Each team consists of one goalkeeper, six defenders, two midfielders and six attackers with the likely configuration displayed in figure 1. The network operates for the lifetime of a single football match with sensing commencing just before, and terminating just after, the match itself. Data outside the First Half and Second Half is to be identified and eliminated from user queries.
- 9-a-side games. Each team consists of one goalkeeper, three defenders, two midfielders and three attackers. As for 15-a-side matches, only data for the First and Second halves should be used in analysis.
- Bangsbo test. The Bangsbo endurance test [1] has a series of levels of increasing intensity. Participants progress through the levels until fatigue forces them to drop out. When the last participant is eliminated, the sensor network terminates.

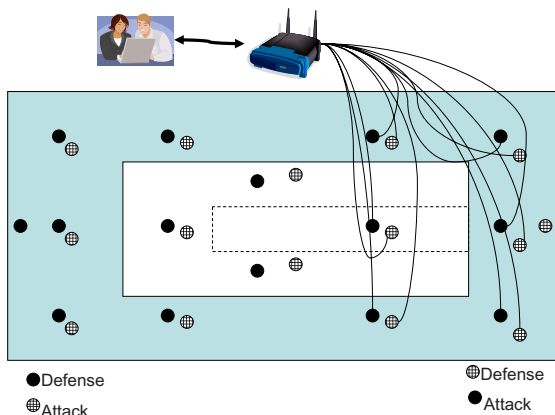


Fig. 1. Network Configurations

From the descriptions above, it is clear that networks may broadcast for set durations or may run for an unknown duration in the case of the Bangsbo test. In addition, the playing surface area differs for all three configurations as illustrated in figure 1. This diagram shows 3 different surface areas: the outside (largest) area is for 15 a side games and will contain 30 nodes for the 15 players on each team; the middle area is for 9-a-side and will contain 18 nodes while the smallest (inner strip) can contain up to 50 nodes as a larger number of players may participate in the training activity in this small area. The user scenario expressed in figure 1 is for an analysis of the 6 attacking players and the defenders who are closely tracking them.

3 Sensor Web Architecture

The concept of the DataSpace system was introduced in [5] as a solution to organisations such as healthcare or sport scientists who have a requirement for large numbers of diverse but interrelated data sources. In this section we describe the major components of the HealthSense DataSpace system and how each component contributes to the information management process.

3.1 Data Capture

The Data Capture Component comprises both data sources and a metabase that is used for understanding the content and semantics of the actual data sources. The key difference between the DataSpace architecture and more traditional distributed architectures is that the data is *subject* oriented, similar to a Data Warehouse. In this DataSpace, sensor data exists in both raw format (binary or textual files) and in an enriched XML format. The raw format is necessary for live queries as the converted files are not available quickly enough for this

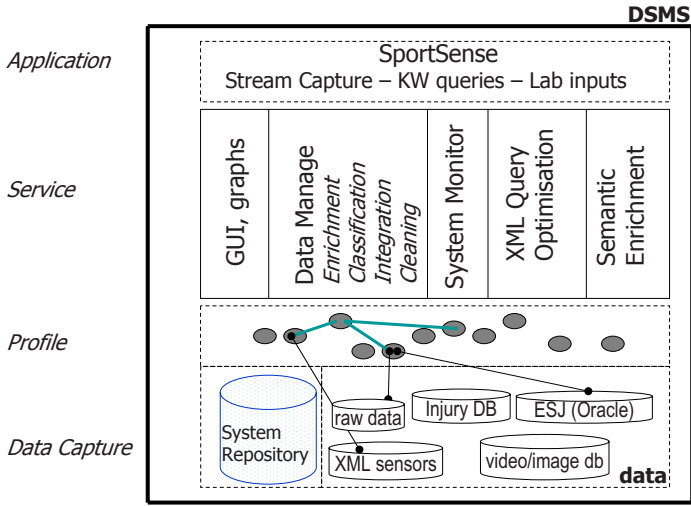


Fig. 2. SportSense DataSpace Architecture

purpose [11]. An Object-relational model (Oracle 10g) is used for creating subject (athlete or patient) profiles and is also used for managing injury data. A separate system is used to store video data while the video metadata is stored in a relational database. Unlike federated database systems that are created from existing application databases, this is a green-field architecture with some level of control to make integration of data easier. However, there will always be situations where unplanned integration will be necessary. For example, combining the output from new sensor devices with previously generated data.

DataSpace Repository. The repository for the HealthSense DataSpace provides the engine for the DataSpace system and has a complex metamodel. As with the DataSpace System itself, the System Repository (or metabase) also adopts a hybrid storage model structure. This is necessary as some of the constructs involved are not suited to traditional storage systems. While a full description of the repository and its Metadata Service form part of a separate body of work [11], we provide a brief description of the major components now.

- **Integrations.** Relationships across separate data sources with semantics for integration.
- **Profiles.** For each user or user type, a profile is created that links the user to specified data sources. It may provide a link to Integration objects where users are managing data from multiple sources.
- **Templates.** Templates are used to describe raw sensor sources. Together with a structural enrichment process, they form XML schemas and are used to populate these schemas with raw data so that they can be queried using

XPath or XQuery [15]. Their goal is to ensure that the service components are never required to change.

- **Contexts.** While template objects provide for the creation of XML data from raw sensor output, this provides only a structural enrichment of the sensor data. With Context objects, it is possible to semantically enrich the file. Contexts provide the necessary background to understand the situation in which each sensor was used. For example, a Heart Rate monitor can be used in a match situation: football, tennis or athletics; or it may be used in testing scenarios such as the Bangsbo test [2].
- **Schemas.** Schema objects are stored for XML databases only. They provide the user with a storage model version of raw sensor data and enable the user to formulate queries. There is a strict one-to-one mapping between templates and schemas.
- **Replicas.** There are many examples of data replication in the DataSpace system and these are modelled in the system repository. For example, the optimiser will create a relational index of an XML database; multimedia metadata is created for video files; XML views are created from object-relational databases for the purpose of sharing data.

3.2 Profile Component

HealthSense is a Web Information System in that web browsers provide the interface to multiple sources of data, and HTML or XML is used as an interface between heterogeneous data collections and users. What connects user types

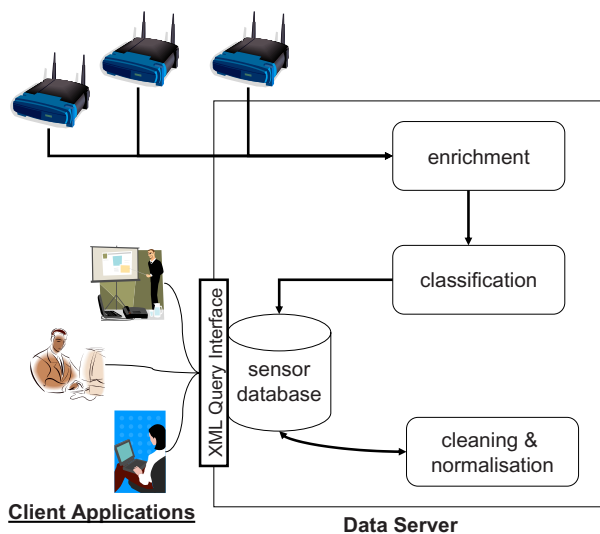


Fig. 3. Data Capture and Management

with different requirements, to one or more data sources are the profiles. Many profiles are simple to construct: a physiotherapist will only query and manage data from a single database (Injury DB) or a knowledge worker who wants to average team heart rate data after each match. However, some of the profiles are more complex: knowledge workers wishing to mine larger data volumes are searching for relationships between training regimes (Electronic Sports journal), maximal heart rate and injuries sustained.

Semantic Enrichment. This service creates the XML version for all sensor streams using a template approach [3] that requires no modification to system components when new sensors are introduced. Initially, raw sensor streams are structurally enhanced to produce basic XML files and subsequently, these files are mined to generate additional semantics for every sensor reading. In all sporting experiments, we apply *state* information to each sensor reading, where a state refers to some interval in either a sporting event (eg. football game or tennis match) or a lab-based training activity.

4 SportsSense System

In this section we describe the infrastructure that facilitates the provision of a data management and query service for a sensor network using the standard web languages XPath and XQuery. While this paper focuses mainly on the cleaning and normalisation of data, we will provide a brief overview of the entire infrastructure.

4.1 Data Enrichment

The role of the enrichment process is to convert the raw sensor data into XML format, providing both structure and additional semantics. This is motivated by the need to use a high level query language rather than write low level primitives every time the user modifies or has a new query requirement. Each sensor device has associated with it a template file (an XML schema document) that facilitates the transformation of the raw data into XML. In previous work [3], we described a system where all sensor streams use an XML template to make themselves readable and queryable by the system. The benefit of this approach is that the system requires no modification when new sensors are introduced to the wireless network. The enriched XML file contains a header section detailing the user information, session parameters that describe the current experiment or activity, and sensor device ID information. The body of the XML file contains the readings recorded by the sensor device.

In figure 4, we have a small extract from an enriched sensor stream which previously generated some header data, followed by a stream of heart rate values and time stamps. Meaningful queries (see §5) are not possible without the descriptive attributes that are added from the sensor's template file. Other than structural markup, additional semantics such as outlier information, rolling averages (to that timestamp) and athlete details are included to facilitate more complex queries.

```

<user>murphy</user>
<session>080503ME_Bangsbo</session>
<sessiontype>Under 14</sessiontype>
<sensorData candidate="candidate">
  <device>HRM</device>
  <startTime>1209826519000</startTime>
  <interval>5000</interval>
  <sections>
    <section name="Params">
      <parameter><key>Version</key>
        <value>106</value>
      </parameter>
      ... (parameter element repeats)
    </section>
    <section name="HRData">
      <measurement offset="0"
state="" stateoffset="" time="1209826519000">
        <reading ordinal="">
          <key>HeartRate</key>
          <raw-value>80</raw-value>
          <outlier-value>80</outlier-value>
          <padded-value>80</padded-value>
          <value>80</value>
          <averages>
            <average>
              <time/>
              <value/>
            </average>
          </averages>
        </reading>
      </measurement>
      ... (measurement element repeats)

```

Fig. 4. An Enriched Sensor Stream

4.2 Classification

The goal of this process is the dynamic classification of sensor streams as different groups or clusters of players (in this sensor network) based on various characteristics, such as age, team, field position and experiment type.

One of the requirements of databases or data warehouses is appropriate classification of all stored objects. This facilitates the user when expressing the query if for example, all related experiments are located in the same section of the database. The benefit of the enrichment process is that sensor data streams now have a number of classifying attributes as shown in figure 4. Currently **user**, **session** and **sessiontype** are used to sort each of the sensor streams as they arrive.

This is an important feature when the system scales to include experiments at a National level. As each Gaelic football club rolls out their own sensor network experiments, it will be necessary to distribute data across multiple sites but have an integration parameter when queries are expressed across distributed information sources. For example, it would be possible to query the results (across the country) for all Bangsbo experiments, involving players under 14 years of age, between a specific range of dates.

4.3 Cleaning and Normalisation

The goal of this process is the calibration of the sensor data so that domain experts can extract and query only appropriate sensor readings. The difficulty with this process is that the network generates data that will corrupt the results of user queries and analyses. This is caused by heart rate monitors sending heart rate data outside the prescribing activities periods. Secondly, the necessary calibration of results requires a number of updates to the sensor streams that can be time consuming using XML databases. Thus, there are two challenges: identifying only relevant sensor data and performing a fast calibration process.

The calibration process involves the use of *states*. By imposing a state value on every sensor reading, we can use the states to categorise and ultimately synchronise sensor streams. All sporting events conform to rigid structures and the training activities employed by the Sports Scientists also adhere to a set structure [1]. Irish Gaelic football has 15 players with the game played over two halves. Scientists require that only heart rates for the duration of the First and Second halves should be used to generate the results of queries. Our algorithm identifies three principle states: First-Half (FH), Half-Time (HT) and Second-Half (SH). An illustration of a typical player heart rate stream over a full game is presented in figure 5. A heart rate sensor begins to monitor as soon as the electrodes touch the skin, and generates a value every 5 seconds. As this takes place in a random and fairly chaotic manner, it is not easy to synchronise sensors at the start of the match. In some cases the monitor is attached 10 minutes before the start and in others up to 40 minutes before the start. However, figure 5 provides some indication as to how the problem was approached as FH and SH states are clearly visible (although only for this participant). Furthermore, all devices were removed within 5 minutes of the end of each match, effectively closing down the sensor network.

The goal of the cleaning process is to remove heart rate values that are not in the First Half (FH) or Second Half (SH) states. For a 15-a-side game, the sports scientists agreed that obtaining a fixed duration of 30 minutes activity for both halves would provide the necessary data to determine the zonal information described in table 1. Thus, we have two fixed functions $T(\text{FH}) = 1800$ and $T(\text{SH}) = 1800$ ($30 * 60$ seconds). It was also agreed that the interval between halves (half-time HT) would be set at 12 minutes, providing $T(\text{HT}) = 720$. These functions will return different values for the 9-a-side networks and also have no meaning for tests such as Bangsbo which has a large number of states, but the principle remains the same, with only the number of functions and their durations changing. It also allows us to parameterise the process where it is clear that playing times were shorter or longer than expected.

The process to clean the data has three core algorithms. The first is `detectHT` which returns the start time for Half Time. Once found this is used to segment the sensor stream in order to reduce the search space for the `detectFH` and `detectSH` algorithms. This step is very important as we discovered that these algorithms return many false positives if the entire stream is searched.

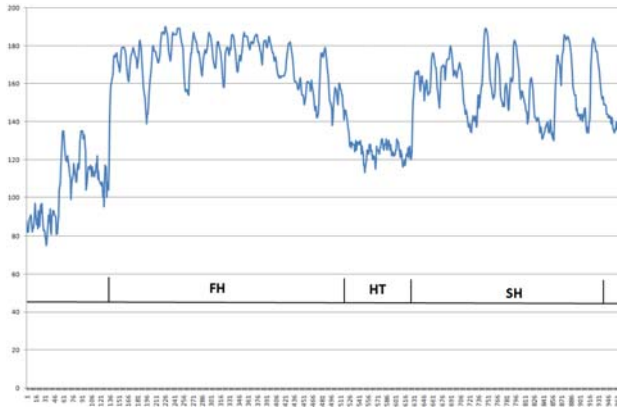


Fig. 5. Player Sensor Data Stream

The `detectHT` algorithm reads from the end of the file and calculates all candidate Half Times by computing rolling averages for every 5-second interval. This must be computed for at least the length of the half, $T(\text{SH})$ and the length of half time, $T(\text{HT})$, and a surplus, $T(\text{surplus})$ to ensure that we always read enough of the data stream. The lowest average heart rate for each 12 minute duration is regarded as half time: the segment of least activity during the search period.

Definition 1. *detectHT*

```
Start from End(Stream)
Calculate each 12 minute average AVG(HT) for  $T(\text{SH})+T(\text{HT})+T(\text{surplus})$ 
for all AVG(i)
  locate smallest
return begin_time for AVG(smallest)
```

The `detectSH` algorithm reads from the end of half time until the end of the sensor stream, and generates each 30 minute average of heart rate values. Upon finding the largest 30 minute average, this is designated as the SH state. A similar process is used to detect the FH state.

Definition 2. *detectSH*

```
Start from End(HT)
Calculate each 30 minute average AVG(SH) until the end of stream
for all AVG(i)
  locate largest
return begin_time for AVG(biggest)
```

As the length for each half and half time will almost never correspond exactly to the predefined durations, there will be gaps between each of the states which are ignored for analysis purposes. However, we present these intervals to the sports scientists in the event that they wish to recalibrate by changing the state durations. For a subsequent iteration, they are used to modify parameters (lengths of playing halves or half time) and create a new range for the sensor values.

We have deliberately kept the algorithms simple to ensure fast query response times, but we have found accuracy to be above 92% with this purely automated approach. Furthermore, the small percentage of streams where we discovered false positives were due to irregular heart rate values. For example, the goal keeper does not have the high activity exhibited by out field players. Additionally, there were a very small number of players who had relatively low activity during playing times, making the true location of half time difficult. Our current approach will employ a Bayesian model to use data from multiple streams (players) to highlight what are impossible Half Time locations.

5 User Queries and Experiments

After enrichment, data is stored in the MonetDB XQuery server [9] where it is then calibrated until ready for user queries. All experiments ran on an Intel Core 2 Duo processor PC with 4GB of RAM running Windows XP professional. The system was implemented using the Sun Java Virtual Machine version 1.6; the MonetDB XQuery server is version 4.30.0. The current size of the dataset is just under 0.5Gb in its XML format in the MonetDb database. All experiments were executed four times with the average of the last three runs times recorded. The first run was treated as a cold run and thus ignored.

5.1 Sample Queries

For the experiments described in this section, our queries focused on the sensor data generated from a single 15-a-side game. On average, each sensor recorded between 1100 and 1200 readings at 5 second intervals over a period of between 95 and 105 minutes. Table 2 provides a list of queries (in English) with the times to compute the results. It provides the overall time for delivery of results to end users. These queries represent the base queries upon which more complex queries are then expressed. For example, to determine those players that exhibit heart rate readings that are, on average, higher than the team average, across a number of matches, it is necessary to build upon the set of basic queries presented in Table 2. While these base queries may appear simplistic, it is not possible to execute these user requests on raw data streams. In other words, without the

Table 2. Query execution times

Query	Time
1 Return all HR values for player number 1 in game time	63 ms
2 Return the Avg HR value of player number 1 in second half game time	94 ms
3 Return the Max HR value of player number 5 in game time	78 ms
4 Return the Avg HR value of Team Scotstown in selected game	141 ms
5 Return Avg HR value for each Scotstown defender in selected game	563 ms
6 Return Max HR value for each Scotstown defender in selected game	531 ms

data management layer, it would require identifying a particular stream (from a potentially large number of data streams), applying some context information to determine the phase "in game time", and writing low-level query primitives to detect values or collect sequences to compute min, max, avg etc. In our system, we can apply XQuery to resolve all user requirements. Additionally, we provide the XQuery expression for each query in Table 3 to demonstrate that even the most basic sensor queries require fairly complex XQuery expressions.

Table 3. Full Query Expressions

Query	Time
1 let \$c := collection('db/GAA_football/Club01') return \$c/healthSense[user[text()='mmccar']/sensorData/sections/section[@name='HRData']/measurement/reading[key[text()='HeartRate']/value/text()]	63 ms
2 let \$c := collection('db/GAA_football/Club01') return fn:avg(\$c/healthSense[user[text()='mmccar']/sensorData/sections/section[@name='HRData']/measurement[@offset>=3600000]/reading[key[text()='HeartRate']/value/text()])	94 ms
3 let \$c := collection('db/GAA_football/Club01') return fn:max(\$c/healthSense[user[text()='bharra']/sensorData/sections/section[@name='HRData']/measurement/reading[key[text()='HeartRate']/value/text()])	78 ms
4 fn:avg(let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player where \$p/user = \$q/Name/Code order by \$q/@Id return if \$q/TeamId="ST" then \$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	141 ms
5 let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player,\$r in \$c//Players where \$p/user = \$q/Name/Code and \$p/session = \$r/Game/Id order by \$q/@Id return if ((\$q/Position/Role="DE" and (\$r/Game/Type="15aside") and (\$q/TeamId="ST")) then fn:avg(\$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	563 ms
6 let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player,\$r in \$c//Players where \$p/user = \$q/Name/Code and \$p/session = \$r/Game/Id order by \$q/@Id return if ((\$q/Position/Role="DE" and (\$r/Game/Type="15aside") and (\$q/TeamId="ST")) then fn:max(\$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	531 ms

5.2 Query Evaluation

The first 3 queries are processed in less than 100ms with the reason being that they query a single player data file (or small XML document). In other words, simple queries based on individuals will always execute quickly, even where the database grows quite large. Even in a database of 0.5Gb, the tree pruning capabilities of MonetDB are quite effective.

It takes a little longer to run Query 4, approximately 150ms, because it calculates the team average heart rate in a specific game and requires access to 15 player files. However, the speed is good as the query locates the match data quickly. This is likely to decrease in speed as large numbers of match data is added. There is currently data from almost 200 matches in the database of

which 80 are 15-a-side matches. We have determined the correct placement and classification of sensor data to improve query response times.

Queries 5 and 6 both require in excess of 500ms to generate results. This is due to the fact that it must compute averages and maximums for all players in a given team.

Table 3 provides the full XQuery expressions for each query to further emphasise the complexity of query expressions and to help explain the query times. Queries 5 and 6 require iterations not needed in other queries. What this table also demonstrates is the level of detail than can be expressed by building an appropriate data management layer that supports standard query languages such as XPath.

6 Related Research

A recent research project that focused on sensor networks in the sporting domain is presented in [4]. They have a similar contextual platform to their experiment in that they treat a single sporting event as the basis for generating the sensor data. Similar to our work, they provide an experimental prototype with fast response times for built-in queries. However, in our approach we provide a high level and flexible query language to allow domain specialists to probe and refine their requirements.

In [16], the authors present a platform for incorporating non-XML sources into an XML system. As with our approach, they use a Description Language to generate the XML representation of data. On the positive side, no conversion of sensor data is necessary as they create a view definition to interpret the raw data. However, they provide only a template system that has not been applied to any domain (instead they provide some use-case descriptions), and no query response times are possible. Our experience with real world data has shown that unexpected data values such as outliers can require system manipulation that affects response times and different networks will provide different problems when trying to synchronise or integrate the sensor data.

In [17], the authors process and query streams of raw sensor data *without* conversion to XML. Their approach is to enrich raw data into semantic streams and process those streams as they are generated. Their usage of constraints on the data streams provides a useful query mechanism with possibilities for optimisation. However, this work is still theoretical and has yet to provide experiments or an indication of query performance.

The researchers in [8] also process raw sensor data without conversion to XML. Here they employ the concept of proximity queries where network nodes monitor and record ‘interesting’ events in their locality. While their results are positive in terms of cost, queries are still at a relatively low level (no common format for query expression), and it is difficult to see how this type of proximity network can be applied in general terms due to the complexity of the technologies involved.

In [7], they provide semantic clusters within their sensor network. This is a similar approach to our approach as we classify related groups of sensor outputs.

They adopt a semi-automated approach and are capable of generating metadata to describe sensors and thus, support query processing. However, their object-oriented approach is likely to lead to problems with interoperability and this could be exacerbated through the lack of common query language. While this can be addressed with a canonical layer (probably using XML) for interoperability, it is likely to have performance related issues. Furthermore, our approach is fully automated with templates used to provide conversion to XML and processes that employ different rule logic for different sensor configurations and contexts.

7 Conclusions

In this paper, we describe our use of sensor networks to measure the work rate intensity during sports training and matches. As the research focused on school-age players, the networks were configured in terms of size, duration and format to assess the physiological impact of participants. Polar's team heart rate monitors were used to generate data and this research describes the data management components that were used to deliver data, through a high-level query interface to domain experts. Data and results presented in this paper were gathered from the sensor networks over a 3-month period in the Summer 2008, and research was jointly carried out by both sports scientists and data management specialists. In our experiments, the slowest query executed in a little over half a second, the system is now in daily use. In effect, sports scientists have the full power of the XQuery language to extract results that are not possible with existing vendor software such as Polar or Garmin where fixed, built-in queries generate tables and graphs.

On the sports science side, current research is focused on providing heterogeneity to the sensor network by incorporating different sensor types which must be integrated with the current data streams. This allows for comparisons between sensor outputs, thus enabling an evaluation of new devices before being deployed outside laboratory environments. From an information management perspective, we are developing a framework that provides the same level of query interface, but for streaming data, so that sports sensor networks can be queried in real time [1]. Furthermore, as all query expressions require somebody with the appropriate IT skills to generate (see Table 3), we are building a parameterised user interface to provide sports scientists with the ability to execute queries without the need for specialist IT involvement.

References

1. Bangsbo, J.: *Fitness Training in Football: A Scientific Approach*, HO+Storm (1994) ISBN 87-983350-7-3
2. Bangsbo, J.: *The Physiology of Soccer - With special reference to Intense intermittent exercise*. *Acta Physiologica Scandinavica* 151(supp. 619) (1994)
3. Camous, F., McCann, D., Roantree, M.: *Capturing Personal Health Data from Wearable Sensors*. In: *International Symposium on Applications and the Internet (SAINT)*, pp. 153–156. IEEE Computer Society Press, Los Alamitos (2008)

4. Devlic, A., Koziuk, M., Horsman, W.: Synthesizing Context for a Sports Domain on a Mobile Device. In: Roggen, D., Lombriser, C., Tröster, G., Kortuem, G., Havinga, P. (eds.) EuroSSC 2008. LNCS, vol. 5279, pp. 206–219. Springer, Heidelberg (2008)
5. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: A New Abstraction for Information Management. *SIGMOD Record* 34(4), 27–33 (2005)
6. Grust, T.: Accelerating XPath Location Steps. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 109–120. ACM Press, New York (2002)
7. Kawashima, H., Hirota, Y., Satake, S., Imai, M.: Met: A Real World Oriented Metadata Management System for Semantic Sensor Networks. In: 3rd International Workshop on Data Management for Sensor Networks (DMSN), pp. 13–18 (2006)
8. Kotidis, Y.: Processing Proximity Queries in Sensor Networks. In: 3rd International Workshop on Data Management for Sensor Networks (DMSN), pp. 1–6 (2006)
9. MonetDB - open source XML database (2008), <http://monetdb.cwi.nl/>
10. Marks, G., Roantree, M.: Metamodel-Based Optimisation of XPath Queries. In: Sexton, A.P. (ed.) BNCOD 2009. LNCS, vol. 5588, pp. 146–157. Springer, Heidelberg (2009)
11. McCann, D., Roantree, M.: A Query Service for Raw Sensor Data. In: Barnaghi, P., et al. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 38–50. Springer, Heidelberg (2009)
12. Polar (2008), <http://www.polar.fi>
13. Reilly, T.: Energetics of high intensity exercise (soccer) with particular reference to fatigue. *Journal of Sports Sciences* 15, 257–263 (1997)
14. Reilly, T., Doran, D.: Science and Gaelic Football: A Review. *Journal of Sports Sciences* 19, 181–193 (2001)
15. Roantree, M., McCann, D., Moyna, N.: Integrating Sensor streams in pHealth Networks. In: 14th Intl. Conf. on Parallel and Distributed Systems, pp. 320–327. IEEE Computer Society Press, Los Alamitos (2008)
16. Rose, K., Malaika, S., Schloss, R.: Virtual XML: A toolbox and use cases for the XML World View. *IBM Systems Journal* 45(2), 411–424 (2006)
17. Whitehouse, K., Zhao, F., Liu, J.: Semantic Streams: a Framework for Composable Semantic Interpretation of Sensor Data. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 5–20. Springer, Heidelberg (2006)

EBC: A Topology Control Algorithm for Achieving High QoS in Sensor Networks

Alfredo Cuzzocrea¹, Dimitrios Katsaros²,
Yannis Manolopoulos³, and Alexis Papadimitriou^{3,*}

¹ ICAR-CNR and University of Calabria, Cosenza, Italy
cuzzocrea@si.deis.unical.it

² Computer and Communication Engineering Department,
University of Thessaly, Volos, Greece
dkatsar@inf.uth.gr

³ Department of Informatics,
Aristotle University, Thessaloniki, Greece
{manolopo, apapadi}@csd.auth.gr

Abstract. A novel approach for achieving high *Quality of Service* (QoS) in sensor networks via *topology control* is introduced and experimentally assessed in this paper. Our approach falls in the broader discipline of *graph structural mining*, and exploits a leading concept initially studied in the context of *Social Network Analysis* (SNA), namely *betweenness*. Particularly, in our research betweenness is applied in terms of a *graph structural mining measure* embedded in the core layer of our proposed topology control algorithm, called *Edge Betweenness Centrality* (EBC). EBC allows us to evaluate relationships between entities of the network (e.g., nodes, edges), and hence identify different roles among them (e.g., brokers, outliers). In turn, deriving knowledge is further exploited to define *raking operators* that look at structural properties of the graph modeling the target sensor network. Based on these amenities, our topology control algorithm is able of providing an “insight” of the graph structure of the network on top which control over information flow, message delivery, latency and energy dissipation among nodes can be easily deployed.

Keywords: Data Mining, Sensor Networks, Topology Control, Graph Structural Mining.

1 Introduction

Recent advances in low-power and short-range-radio technology arisen during last few years have enabled a rapid development of *Wireless Sensor Networks* (WSN). The range of applicability of WSN is very wide, and spans from environmental sensor networks monitoring (environmental) parameters, such as temperature and humidity, to industrial control robotics, from disaster prevention systems to emergency management systems, and so forth. Sensors are tiny,

* Author names are in alphabetical order.

usually battery-operated devices with radio and computing capabilities, which are used to cooperatively monitor physical or environmental conditions.

As regards research issues of sensor networks, several efforts have been done by both the academic and industrial research community, mainly in the context of *routing algorithms* [11, 13], *network coverage aspects* [15, 27], *storage issues* [18, 25] and *topology control* [16, 24]. The common denominator of all these efforts is represented by the goal of *maximizing energy conservation* across the network, in order to gain efficacy and efficiency, as maximizing energy conservation corresponds to *maximizing network lifetime*. For instance, as regards specific data management issues over sensor networks [4], maximizing energy conservation means that *multi-step* maintenance and query algorithms can be executed over the target sensor network, thus involving in more effective data management capabilities rather than the case of *single-step* algorithms. Another motivation of the need for energy conservation in sensor networks relies on inherent technological properties of sensors. In fact, sensors are unlikely to be recharged, especially since they may be deployed in unreachable terrains, or, in some cases, they may be disposed after the monitoring application running over the target network ends its execution.

In order to reduce energy consumption, *topology control algorithms* have been proposed in literature [10, 12, 16, 17, 19, 22, 23, 24, 28, 29, 30]. The final goal of these algorithms consists in reasoning-over and managing the topology of the graph modeling the target sensor network in order to reduce energy consumption as much as possible hence increase network lifetime accordingly. A different line of research appeared recently proposes driving sensor network topology control in terms of *Quality of Service* (QoS) requirements [17] over the target sensor network itself. Several QoS-based requirements have been designed and developed in this context, depending on the particular application scenario ranging from real-time video and content provisioning to time-critical control systems, and so forth (see [17] for a complete survey of typical case studies). Given a set of nodes performing a specific task which is critical for the target sensor network application (e.g., sink nodes in environmental sensor networks), the basic idea behind topology control algorithms is to select from the target network appropriate *logical neighbors* of the former nodes, namely a subset of *physical neighbors* of the former nodes that can be used to perform application-specific procedures (e.g., message transmission) without the need of involving the rest of physical neighbors during the execution of these procedures. QoS-based topology control algorithms select the suitable set of logical neighbors such that input QoS requirements can be satisfied.

Inspired by motivations above, in this paper we investigate the problem of QoS-based topology control over *homogenous WSN*. Given (i) a set of wireless nodes in a plane such that nodes have the same transmitting power and bandwidth capacity, and (ii) QoS requirements between node pairs, our problem consists in finding a network topology that can simultaneously meet the input QoS requirements and minimize the maximal power utilization ratio of nodes. In particular, in our research QoS requirements are modeled in terms of simple-yet-effective *node*

connectivity, so that message transmission can be ensured (while node connectivity can be preserved in order to ensure correct message delivery), and network lifetime can be increased as much as possible accordingly. In this scenario, *avoidance of hotspots* also needs to be carefully considered. Therefore, *adaptive tasks* that depend on the current logical neighbor seem to play the role of most promising strategy to be investigated in order to avoid fast battery depletion.

Looking at deeper details, in our research we propose an innovative *weighted, bidirectional topology control algorithm*, called *Edge Betweenness Centrality* (EBC), and experimentally evaluate such algorithm against a state-of-the-art topology control algorithm, namely *Gabriel Graph* (GG) [8]. Fundamentals of our approach can be found in the conceptual basis drawn by several *centrality measures* that have been proposed in order to model and evaluate the *importance* of a node in a network [3, 9]. These measures have been initially applied in the context of *Social Network Analysis* (SNA), and later to other areas as well, such as *biological networks* [31].

Freeman [6, 7] defines the *betweenness* of a node as a possible centrality measure for detecting the importance of that node within the target network, thus achieving the fundamental concept of *betweenness centrality*. This concept founds on the property stating that vertices that occur on many shortest paths between other vertices have *higher* betweenness than those with lower occurrences. *Closeness centrality* [7] pinpoints vertices that tend to have short geodesic distances from other vertices with in the network. In network analysis, closeness is preferred over shortest-path length, as closeness gives higher values to more central vertices [7]. Finally, *Eigenvector centrality* [1] assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes provide to the global score of the actual node a higher contribution rather than the one provided by connections to low-scoring nodes. For instance, Google's *PageRank* [21] is a variant of the Eigenvector centrality measure. Our research focuses on a meaningful variation of the betweenness centrality concept, namely *edge betweenness centrality* [9, 20], and its application to the leading context of sensor networks.

Summarizing, the contributions of this paper are the following:

- an innovative weighted, bidirectional topology control algorithm, EBC, and its application to the leading context of sensor networks;
- a comprehensive experimental evaluation of algorithm EBC, and its comparison with a state-of-the-art topology control algorithm, GG, on top of the well-known simulation environment *JSim* [26];
- critical analysis and discussion on performance of the two comparison topology-control algorithms, EBC and GG.

The rest of the paper is organized as follows. In Section 2 we discuss related work on topology control algorithms over networks. Section 3 describes in detail algorithm EBC. Section 4 focuses on the comparison algorithm GG. Section 5 is devoted to the experimental evaluation and analysis of the two comparison topology control algorithms, EBC and GG. Finally, Section 6 contains conclusions and future work of our research.

2 Related Work

There exists considerable related work addressing topology control issues over networks, even focalizing on QoS-based topology control. As regards studies on topology management for energy conservation in networks, it has been demonstrated that both powering off redundant nodes and lowering radio power while maintaining node connections can contribute to efficient power saving. In light of this assumption, Shen et al. [24] introduce algorithm *Local Shortest Path* (LSP). In the LSP approach, each node makes use of link weights in order to compute the shortest paths between itself and neighboring nodes. Then, all second nodes on these shortest paths are selected as logical neighbors. The final step of algorithm LSP involves in adjusting the power transmission of so-selected logical nodes to save energy.

Li et al. [19] instead propose algorithm *Local Minimum Spanning Tree* (LMST), which computes a “power-reduced” network topology by constructing a minimum spanning tree over the network in a fully-distributed manner. The aim of this approach relies in the evidence that the power-reduced network is less energy-consuming than the original network.

EasiTPQ [17] is another QoS-based topology control algorithm. EasiTPQ founds on the assumption that each node in the network has different functionalities during data transmission, e.g. some nodes bear more data relay tasks whereas some other nodes only transmit data generated by themselves. In order to achieve the desired QoS, EasiTPQ schedules as active nodes that are more involved in relaying data tasks rather than generating data flows.

Wattenhofer et al. [29] propose a simple-yet-effective distributed algorithm according to which each node makes *local decisions* about its transmission power, such that these local decisions then collectively guarantee *global connectivity* of the network. Specifically, based on directional information, a node grows its transmission power until it finds a neighbor node in every possible direction. The resulting network topology increases network lifetime by reducing transmission power, and, in turn, even traffic interference, thanks to the deriving availability of low-degree nodes. Huang et al. [12] further extend [29] to the case of using *directional antennas*.

Ramanathan and Rosales-Hain [23] describe a *centralized spanning tree* algorithm for building connected and bi-connected networks with the goal of minimizing the maximum transmission power for each node. Two optimal, centralized algorithms, namely CONNECT and BICONN-AUGMENT, are proposed for the case of static networks. Both are greedy algorithms, and resemble Kruskal’s minimum cost spanning tree algorithm [14]. For the case of hoc wireless networks, two distributed heuristics have been proposed, namely LINT and LILT. However, these heuristics do not guarantee network connectivity.

Finally, Jia et al. [30] focus the attention on the problem of determining a network topology able to meet input QoS requirements in terms of end-to-end delay and bandwidth. The proposed scheme adopts an optimization criterion whose goal is to minimize the maximum per-node power consumption. In [30],

authors demonstrate that, when network traffic is “splittable”, a sub-optimal solution can be achieved by means of linear programming techniques.

3 Edge Betweenness Centrality: A Novel Topology Control Algorithm for Sensor Networks

During past years, *vertex betweenness* has been studied in the vest of a measure of the centrality and influence of nodes in networks [6, 7]. Given a node v_i , vertex betweenness is defined as the number of shortest paths between pairs of nodes that run through v_i . Vertex betweenness is a measure of the influence of a node over the information flow among nodes of the network, especially in scenarios such that information flowing over the target network primarily follows shortest available paths.

In order to compute betweenness centrality, Brandes [2] proposes an efficient *backwards algorithm* which starts from leaf nodes of a tree of shortest paths and progressively accumulates the leaf-nodes’ betweenness values moving back towards the root node of the tree.

Girvan-Newman algorithm [9] extends the definition of betweenness centrality from network vertices to network edges, via introducing the concept of *Edge Betweenness* (EB). Let $G = \langle V, E \rangle$ be a connected undirected graph, and v_i and v_j two nodes in G , respectively. Let $\sigma_{v_i v_j}$ denote the number of shortest paths between nodes v_i and v_j . Let $\sigma_{v_i v_j}(e)$ denote the number of shortest paths between v_i and v_j which go through $e \in E$. Betweenness centrality of an edge $e \in V$, denoted by $EB(e)$, is defined as follows:

$$EB(e) = \sum_{v_i \in V} \sum_{v_j \in V} \frac{\sigma_{v_i v_j}(e)}{\sigma_{v_i v_j}} \quad (1)$$

In its original implementation [20], which focuses on unweighted, undirected networks, EB analysis makes use of algorithm *Breadth-First Search* (BFS). Girvan-Newman algorithm [9] works in the opposite way. Instead of trying to construct a measure that determines edges that are the “most central” for network communities, it focuses on edges that are the “least central” for network communities, i.e. edges that are “most between” for network communities. Communities are detected by progressively removing edges from the original graph, rather than by adding the strongest edges to an initially empty network. In our research, we do not use the centrality measure to find communities but instead to select the most important edges, energy-wise, to propagate messages.

Specifically, steps that are used to compute the edge betweenness centrality index are the following:

1. compute shortest paths through the network by means of Dijkstra’s algorithm [5];
2. for each edge, compute the edge betweenness centrality index like in [20], but instead of un-weighted edges use the *average energy* of the two connecting nodes as edge weight.

Based on the edge betweenness centrality index, our algorithm EBC selects logical neighbors of actual node based on the following rules:

- for each node, logical neighbors must cover the 2-hop node neighborhood;
- 1-hop neighbors with the highest-scoring betweenness centrality index are selected.

Moreover, in order to avoid hotspots, our algorithm recalculates the edge betweenness centrality index based on the corresponding energy levels of each node, therefore selecting different edges to be part of the logical neighborhood of each node.

4 Gabriel Graph: A State-of-the-Art Topology Control Method for Networks

Gabriel Graph has been introduced by Gabriel and Sokal in [8]. Formally, given a graph $G = \langle V, E \rangle$ and two vertices v_1 and v_2 in V , we say that v_1 and v_2 are *adjacent* if the closed disc of diameter v_1v_2 does not contain other vertices of V . In the context of sensor networks, we extend the basic adjacency concept above and we say that a sensor node s_i is *connected* with a sensor node s_j , who lies within the s_i 's transmission range, if there not exist another node s_k which is contained by the closed disc of diameter $s_i s_j$. This simple-yet-effective method is used by algorithm GG to find logical neighbors of a given sensor node.

In more detail, in our JSim-based experimental framework, logical neighbors of a given sensor node are found by algorithm GG according to the following steps:

1. each sensor node broadcasts its location – at the end, every node in the sensor network knows its neighbors and their locations;
2. each sensor node s_i determines its logical neighbor set L_i by computing the closed discs of diameters equal to the distance between the location of s_i and each other physical node belonging to the s_i 's physical neighborhood set P_i – for each physical neighbor s_j in P_i , if the disc of diameter $s_i s_j$ does not contain other physical neighbors of P_i then s_j becomes a logical neighbor of s_i .

5 Experimental Evaluation and Analysis

In this Section, we present the experimental evaluation of algorithm EBC in comparison with algorithm GG, which can be reasonably considered a state-of-the-art result in topology control over networks.

5.1 Simulation Model

In our experimental framework, we have developed a simulation model based on JSim, a well-known Java-based simulation environment for numerical analysis [26]. In particular, in our simulation environment, the AODV routing protocol

Table 1. Simulation parameters

Parameter	Values
sensor node number	500, 750, 1000
terrain size	400 x 400
radio range	14m, 17m
initial energy charge	10 Joules
transmission energy	0.001 Joules
wireless bandwidth	2 Mbps

is deployed within the reference WSN [REF]. Also, we use IEEE 802.11 as the MAC protocol and the free space model as the radio propagation model. Wireless bandwidth is assumed to be 2 Mbps.

We performed a large number of experiments on top of various sensor network topologies, and by ranging several experimental parameters, but for the interest of space, here we present a subset of our experimental results. Table 1 summarizes the simulation parameters.

5.2 Experimental Results

As stated in previous sections, topology control algorithms over sensor networks try to minimize the energy consumption of nodes by transmitting data to a subset of a node's physical neighbors. Therefore, given the actual node, the first step deals with the issue of finding node's physical neighbors. Then, topology control algorithms are applied in order to select the subset of logical neighbors that can propagate messages throughout the network without any data loss, neither involving all the effective physical neighbors.

Our experimental analysis focuses on the comparison between algorithms EBC and GG in terms of logical neighbors found and energy consumption that is needed to propagate messages through logical neighbors. For each algorithm, we also analyze the impact of a change in network density on algorithm's performance.

Figure 1 shows the overall number of physical neighbors that exist in the network for 500, 750 and 1000 nodes, respectively. The increase in the number of physical neighbors is due to the increase in the sensor transmission radius from 14 to 17 meters. This means that each sensor node can communicate with nodes that exist in its wider vicinity. For a radius of 14m, the number of physical neighbors are 1298, 2640 and 4488, respectively. For a radius of 17m, we instead have: 1958, 3984, and 6797.

Figure 2 illustrates the average number of physical neighbors of each node in the network, for different size of the sensor network. In the first case, i.e. a network with 500 nodes, the average number of physical nodes per-sensor-node is 2.4 for a radius of 14m and 3.7 for a radius of 17m. The respective numbers for a network with 750 nodes are: 3.5 (14m radius) and 5.3 (17m radius). Finally, for a network with 1000 nodes, retrieved numbers are: 4.4 (14m radius) and 6.7

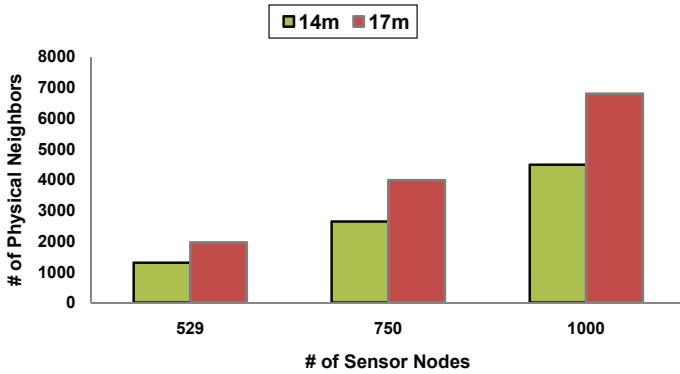


Fig. 1. Number of physical neighbors

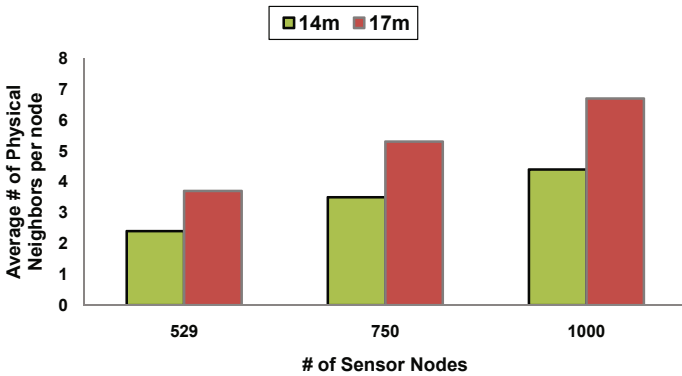


Fig. 2. Average number of physical neighbors per-sensor-node

(17m radius). Notice that in all the cases retrieved numbers are the same for both algorithms EBC and GG since they are not applied to the initial step that finds the physical neighbors of each node.

Moving the attention on the proper experimental comparison of the two investigated topology-control algorithms (i.e., EBC and GG), Figure 3 shows the overall number of logical neighbors found after each algorithm has been applied to each network setting with different size (500, 750 and 1000 nodes) when the radius is set to 14m. As shown in the Figure, starting from an initial number of physical neighbors found equal to 1086 (500 nodes), algorithm GG finds 1298 logical neighbors (750 nodes) while algorithm EBC finds a smaller subset of 752 logical neighbors (750 nodes). This difference increases as the number of sensor nodes in the network increases. For 1000 nodes, algorithm GG finds 3742 logical neighbors, whereas algorithm EBC 1513 logical neighbors only.

Figure 4 shows instead the performance of algorithms EBC and GG in terms of average logical neighbors found per-sensor-node, still with a radius equals to

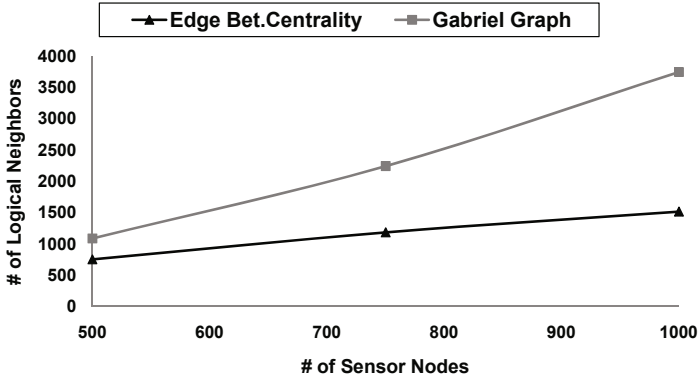


Fig. 3. Number of logical neighbors found (radius = 14m)

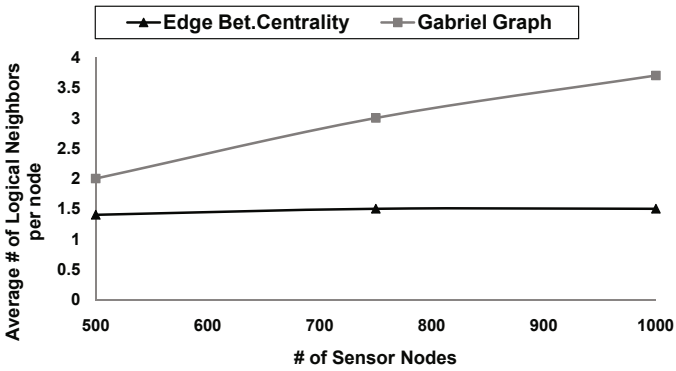


Fig. 4. Average number of logical neighbors per-sensor-node (radius = 14m)

14m. As clearly follows from Figure 4, algorithm EBC delivers *about the same* average number of logical neighbors per-sensor-node, i.e. about 1.5, irrespectively of the size of the sensor network. On the other hand, algorithm GG does not perform as well, since the average number of logical neighbors per-sensor-node ranges from 2 (500 nodes) up to 3.7 (1000 nodes).

Figure 5 shows the results for the same experiment when the radius is set to 17m. As shown in the Figure, when radius increases the difference between the two algorithms' performance is even more noticeable. In fact, the number of logical neighbors found by algorithm GG ranges from 1639 (500 nodes) to 5648 (1000 nodes). The respective numbers for algorithm EBC range instead from 1014 (500 nodes) to 2052 (1000 nodes). Therefore, it clearly follows that EBC outperforms GG even under this experimental analysis perspective.

Figure 6 confirms to us the superiority of algorithm EBC over algorithm GG in terms of the average number of logical neighbors found per-sensor-node, still with a radius equals to 17m. It should be notice again that algorithm EBC

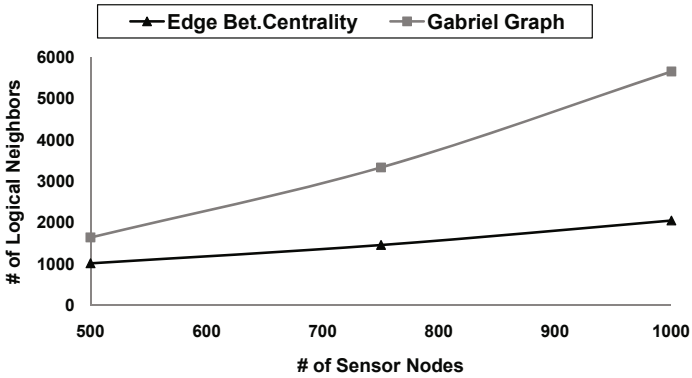


Fig. 5. Number of logical neighbors found (radius = 17m)

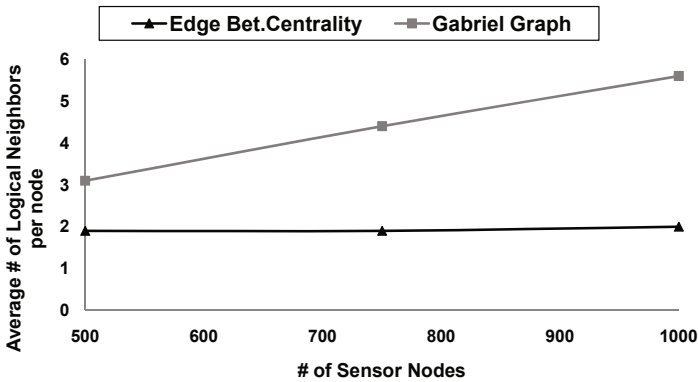


Fig. 6. Average number of logical neighbors per-sensor-node (radius = 17m)

remains practically insensitive to the increase in the number of sensor nodes and provides an average number of 2 logical neighbors per-sensor-node throughout the simulation. On the other hand, algorithm GG performs poorly with an average number of logical neighbors found per-sensor-node ranging from 3.1 (500 nodes) to 5.6 (1000 nodes).

Looking at energy consumption minimization, the main goal of topology control algorithms, Figure 7 shows the energy consumption per-node needed to propagate a message to logical neighbors, when the radius is set to 14m. Again, algorithm EBC requires an almost unchanged amount of energy to this goal, i.e. about 0.0015 Joules, whereas algorithm GG requires an amount of energy ranging from 0.0020 (500 nodes) to 0.0037 (1000 nodes) Joules to perform the same operation.

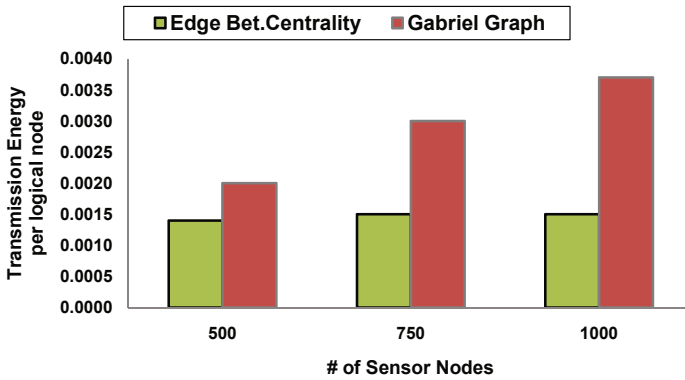


Fig. 7. Transmission energy consumption per-node (radius = 14m)

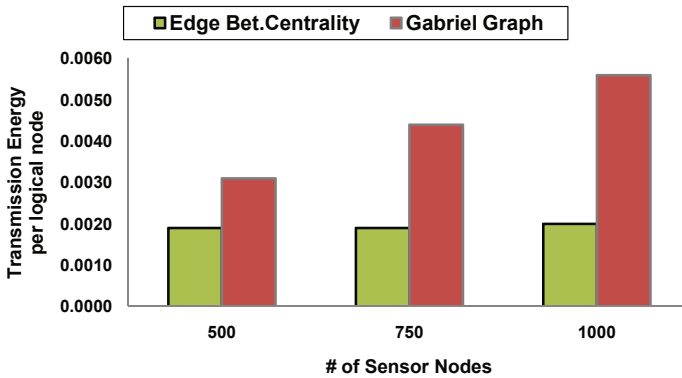


Fig. 8. Transmission energy consumption per-node (radius = 17m)

Finally, Figure 8 shows the results for the same experiment when the radius is set to 17m. Even in this experimental analysis, algorithm EBC outperforms algorithm GG with a transmission energy consumption per-node equals to of 0.002 Joules. Indeed, algorithm GG significantly increases the energy requirement by ranging from 0.0031 (500 nodes) to 0.0056 (1000 nodes) Joules.

6 Conclusions and Future Work

Betweenness is a centrality measure for networks that has been initially studied in the context of SNA. This measure states that vertices that occur on

many shortest paths between other vertices have higher betweenness than those with lower occurrences. Therefore, nodes with high betweenness are selected as nodes able to control the overall information flow within the network. Topology control algorithms aim at providing high QoS by maximizing network lifetime and ensuring message delivery. Inspired by these motivations, in this paper we have proposed a novel topology control algorithm for sensor networks, EBC, which exploits the edge betweenness centrality concept to ensure high QoS throughout the network. Also, we performed a comprehensive campaign of experiments where we compared the performance of algorithm EBC with the performance of algorithm GG, a state-of-the-art result in topology control over networks, under several perspectives of analysis. Experimental results have clearly demonstrated the superiority of algorithm EBC over algorithm GG, in terms of both logical neighbors found and amount of transmission energy consumption.

As future work, we plan to devise alternative centrality measures for networks, looking at the wide literature available on the topic, and experimentally compare these novel measures to edge betweenness centrality. Apart from number of logical neighbors found, transmission energy consumption and scalability, which have been investigated in this paper, in the future experimental analysis we will focus on other interesting experimental parameters that need more research efforts, such as message latency and message delivery.

References

1. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* 2(1), 113–120 (1972)
2. Brandes, U.: A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* 25, 163–177 (2001)
3. Brandes, U.: On variants of shortest-path betweenness centrality and their generic computation. *Social Networks* 30(2), 136–145 (2008)
4. Cuzzocrea, A.: *Intelligent techniques for warehousing and mining sensor network data*. IGI Global (2009)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1), 269–271 (1959)
6. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* 40(1), 35–41 (1977)
7. Freeman, L.C.: Centrality in social networks: Conceptual clarification. *Social Networks* 1(3), 215–239 (1979)
8. Gabriel, R.K., Sokal, R.R.: A new statistical approach to geographic variation analysis. *Systematic Zoology* 18(3), 259–278 (1969)
9. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U S A* 99(12), 7821–7826 (2002)
10. Hackmann, G., Chipara, O., Lu, C.: Robust topology control for indoor wireless sensor networks. In: *SenSys 2008: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 57–70. ACM, New York (2008)

11. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: HICSS 2000: Proceedings of the 33rd Hawaii International Conference on System Sciences, Washington, DC, USA, vol. 8, p. 8020. IEEE Computer Society, Los Alamitos (2000)
12. Huang, Z., chung Shen, C., Srisathapornphat, C., Jaikaeo, C.: Topology control for ad hoc networks with directional antennas. In: Proc. IEEE Int. Conference on Computer Communications and Networks, pp. 16–21 (2002)
13. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.* 11(1), 2–16 (2003)
14. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7(1), 48–50 (1956)
15. Liu, B., Brass, P., Dousse, O., Nain, P., Towsley, D.: Mobility improves coverage of sensor networks. In: *MobiHoc 2005: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 300–308. ACM, New York (2005)
16. Liu, J., Li, B.: Distributed topology control in wireless sensor networks with asymmetric links (2003)
17. Liu, W., Cui, L., Niu, X., Liu, W.: Easitpq: Qos-based topology control in wireless sensor network. *J. Signal Process. Syst.* 51(2), 173–181 (2008)
18. Mathur, G., Desnoyers, P., Ganesan, D., Shenoy, P.: Ultra-low power data storage for sensor networks. In: *IPSN 2006: Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 374–381. ACM, New York (2006)
19. Li, N., Hou, J.C., Sha, L.: Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications* 4(3), 1195–1206 (2005)
20. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks (August 2003)
21. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
22. Pan, J., Hou, Y.T., Cai, L., Shi, Y., Shen, S.X.: Topology control for wireless sensor networks. In: *MobiCom 2003: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 286–299. ACM, New York (2003)
23. Ramanathan, R., Rosales-hain, R.: Topology control of multihop wireless networks using transmit power adjustment, pp. 404–413 (2000)
24. Shen, Y., Cai, Y., Xu, X.: A shortest-path-based topology control algorithm in wireless multihop networks. *SIGCOMM Comput. Commun. Rev.* 37(5), 29–38 (2007)
25. Sheng, B., Li, Q., Mao, W.: Data storage placement in sensor networks. In: *MobiHoc 2006: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 344–355. ACM, New York (2006)
26. Sobeih, A., Hou, J.C., Kung, L.-C., Li, N., Zhang, H., Chen, W.-P., Tyan, H.-Y., Lim, H.: J-Sim: A simulation and emulation environment for wireless sensor networks. *IEEE Wireless Communications Magazine* 13(4), 104–119 (2006)
27. Thai, M.T., Wang, F., Du, D.H., Jia, X.: Coverage problems in wireless sensor networks designs and analysis. *Int. J. Sen. Netw.* 3(3), 191–200 (2008)

28. Tseng, Y.-C., chee Tseng, Y., Chang, Y.-N., hour Tzeng, B.: Energy-efficient topology control for wireless ad hoc sensor networks (2002)
29. Wattenhofer, R., Li, L., Bahl, P., min Wang, Y.: Distributed topology control for power efficient operation in multihop wireless ad hoc networks, pp. 1388–1397 (2001)
30. Jia, D.L.X., Du, D.-Z.: Qos topology control in ad hoc wireless networks. In: IEEE Infocom 2004 (2004)
31. Yoon, J., Blumer, A., Lee, K.: An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics* 22(24), 3106–3108 (2006)

Self-organization and Local Learning Methods for Improving the Applicability and Efficiency of Data-Centric Sensor Networks^{*}

Gabriele Monti and Gianluca Moro

DEIS - University of Bologna,
Via Venezia, 52,
I-47023, Cesena (FC), Italy
{gabriele.monti4,gianluca.moro}@unibo.it

Abstract. In data-centric sensor networks each device is like a minimal computer with cpu and memory able to sense, manage and transmit data performing in-network processing by means of insertions, querying and multi-hop routings. Saving energy is one of the most important goals, therefore radio transmissions, which are the most expensive operations, should be limited by optimizing the number of routings. Moreover the network traffic should be balanced among nodes in order to avoid premature discharge of some devices and then network partitions. In this paper we present a fully decentralized infrastructure able to self-organize fully functional data centric sensor networks from local interactions and learning among devices. Differently from existing solutions, our proposal does not require complex devices that need global information or external help from systems, such as the Global Positioning System (GPS), which works only outdoor with a precision and an efficacy both limited by weather conditions and obstacles. Our solution can be applied to a wider number of scenarios, including mesh networks and wireless community networks. The local learning occurs by exploiting implicit cost-free overhearing at sensors. The work reports an extensive number of comparative experiments, using several distributions of sensors and data, with a well-know competitor solution in literature, showing that an approach fully based on self-organization is more efficient than traditional solutions depending on GPS.

1 Introduction

Self-organization is becoming a promising paradigm to cope with complex systems and to reduce their costs, in fact it leads, in general, to properties like the self-configuration, self-administration and self-healing etc., namely to systems that work without, or drastically limiting, the human interventions. Application scenarios where it can be inconvenient or unfeasible to set up a system by

^{*} Work partially funded by the european project DORII: Deployment of Remote Instrumentation Infrastructure Grant agreement no. 213110.

configuring or implementing every single component are also emerging in large wireless ad-hoc networks.

Examples of ad-hoc networks are mesh networks, sensors networks and wireless community networks (i.e. static or quasi-static networks), while vehicular networks are an interesting example of MANET (mobile ad-hoc networks). Compared to wired networks, wireless networks have unique characteristics. In wireless networks, node failures may cause frequent network topology changes, which instead are rare in wired networks. In contrast to the stable link capacity of wired networks, wireless link capacity continually varies because of the impacts from transmission power, receiver sensitivity and interference. Additionally, wireless sensor networks have strong power restrictions and bandwidth limitations.

In this paper we focus on self-organizing sensor networks, though our work is easily suitable for all static (or quasi static) wireless ad-hoc networks, as we highlight in some rows. Several kinds of sensor applications have been developed in recent years. In some applications, a large volumes of data or events are continuously collected, aggregated/synthesized and stored by sensors for in-network processing. Data-Centric Storage (DCS) scheme emerged from the sensor network literature as the most efficient one for storing and processing data directly within a sensor network. This kind of networks are possible thanks to a new generation of sensors equipped with memory for storing data and processors for executing moderately demanding algorithms. In other words such sensors are similar to minimal computers but with more restrictions, hence the solution presented in this paper is also suitable to the mentioned above networks composed by more powerful devices.

In DCS, events are placed according to their event types, which refers to predefined attribute's values (temperature and pressure, for instance). Hence data or events can be named by attributes and logically represented as relations in distributed databases [7] [1] [4].

In this paper we present a new solution based on a local learning method that improves the performance of W-Grid infrastructure [10] [11] [9] [8], both in terms of routings up to 12% and of traffic balancing, without affecting energy consumptions; we highlight that it is difficult to gain both performance on routing and on balancing just because there is a trade-off between them. The solution does not require GPS because each device receives a virtual coordinate reflecting its local connectivity with other neighbour devices and each of them uses this information to perform routings and to forward exact match query, namely a query to search a single exact data. This means a greater applicability than existing solution based on GPS. The work also present how the infrastructure manages range queries, which are more complex searches involving two or more attributes/dimensions. The in-network data management occurs spontaneously by observing that each device receives a set of multiple unique virtual coordinates, each of which represents also a portion of the data indexing space for which a device is assigned the management responsibility.

Section 2 describes the related works, in Section 3 we briefly present the main features of the infrastructure. Section 4 illustrates the management of

data and queries. Section 5 introduces the cost-free local learning capability. Section 6 describes an extensive number of comparative experiments according to several scenarios; the performances are compared with the preceding version of W-Grid and with a well-known competitor solution in literature that requires GPS. Section 7 reports final considerations.

2 Related Works

Routing is necessary whenever a data sensed (generated) must be transmitted elsewhere in the network, including an external machine, proactively or reactively according to periodic tasks or queries submitted to the network system.

As stated before, we do not consider sensor networks which simply transmit data externally at a remote base station, we focus on advances wireless sensor networks in which data or events are kept at sensors, are indexed by attributes and represented as relations in a virtual distributed database. For instance in [4,15], data generated at a node is assumed to be stored at the same node, and queries are either flooded throughout the network [4].

In a GHT [13], data is hashed by name to a location within the network, enabling highly efficient rendezvous. GHTs are built upon the GPSR [5] protocol and leverage some interesting properties of that protocol, such as the ability to route to a sensors nearest to a given location, together with some of its limits, such as the risk of dead ends. Dead end problems, especially under low density environment or scenarios with obstacles or holes, are caused by the inherent greedy nature of the algorithm that can lead to situation in which a packet gets stuck at a local optimal sensors that appears closer to the destination than any of its known neighbors. In order to solve this flaw, correction methods such as perimeter routing, that tries to exploit the right hand rule, have been implemented. However, some packet losses still remain and furthermore using perimeter routing causes loss of efficiency both in terms of average path length and of energy consumption. Besides, another limitation of geographic routing is that it needs sensors to know their physical position adding localization costs to the system. In DIFS [3], Greenstein et al. have designed a spatially distributed index to facilitate range searches over attributes.

Our solution is more similar to the multi-dimensional distributed indexing method for sensor networks developed in [6] and [14], but differently from our approach they require nodes to be aware of their physical location and of network perimeter; moreover they employ GPSR for the physical routing. GPSR routing performances are heavily affected by network topology (e.g nodes density or obstacles) and it cannot work in indoor environments since it relies on GPS. Our solution behaves like a multi-dimensional distributed index, but its indexing feature is cross-layered with routing, meaning that no physical position nor any external routing protocol is necessary, routing information is given by the index itself.

In [6] and [14] data space partitions, whose splitting method derives from [12], follow the physical positions of nodes, instead of the distribution of data. The consequence is that the storage load per node is, in general, unbalanced, because

it depends on the physical network topology; this leads to an unbalanced number of routings among nodes, particularly with not random data as shown by the experiments, and consequently to a rapid network break-up caused by premature turning off of most loaded sensors. In W-Grid the storage load balancing has been achieved thanks to two key points: (i) the multi-dimensional data space partitions occur according to the actual data distribution and (ii) each partition has the same maximum bucket size. Another key difference is that data partitions in [6] and [14] are disjoint, while in W-Grid they are nested. The main difference between [6] and [14] is that the latter requires a fewer number of sensors with GPS.

3 W-Grid

From now on, in this paper we will use the term nodes and sensors interchangeably. The main idea is to map sensors on a binary tree so that the resulting coordinate space reflects the underlying connectivity among them. Basically we aim to set parent-child relationships to the sensors which can sense each other, in this way we are always able to route messages, in the worst cases simply following the paths indicated by the tree structure. Using virtual coordinates that do not try to approximate node's geographic position we eliminate any risk of dead-ends. Basically W-Grid can be viewed as a binary tree index cross-layering both routing and data management features in that, (1) by implicitly generating coordinates and relations among nodes allows efficient message routing and, at the same time, (2) the coordinates determine a data indexing space partition for the management of multi-dimensional data. Each node has one or more virtual coordinates on which the order relation is defined and through which the routing occurs, and at the same time each virtual coordinate represents a portion of the data indexing space for which a device is assigned the management responsibility. W-Grid virtual coordinates are generated on a one-dimensional space and the devices do not need to have knowledge of their physical location. Thus, differently from algorithms based on geographic routing (see section 2), W-Grid routing is not affected by dead-ends. Since in sensor networks the most important operations are data gathering and querying it is necessary to guarantee the best efficiency during these tasks.

3.1 Generation of Virtual Coordinates

When a device, let us say d turns on for the first time, it starts a wireless channel scan (beaconing) searching for any existing W-Grid network to join (namely any neighbor device that already holds W-Grid virtual coordinates). If none W-Grid network is discovered, d creates a brand new virtual space coordinate and elects itself as root by getting the virtual coordinate “*”¹. On the contrary, if beaconing returns one or more devices which hold already a W-Grid coordinate, n will join the existing network by getting a virtual coordinate.

¹ It is conventional to label “*” the root node.

Coordinate Setup. Whenever a node needs a new W-Grid coordinate, an existing one must be split. A new coordinate is given by an already participating node d_g , and we say that its coordinate c is split by concatenating a 0 or a 1 to it. The result of a split to c will be $c' = c+1$ and $c'' = c+0$. Then, one of the new coordinates is assigned to the joining node, while the other one is kept by the giving node. No more splits can be performed on the original coordinate c since this would generate duplicates. In order to guarantee coordinates' univocity even in case of simultaneous requests, each asking node must be acknowledged by the giving one d_g . Thus, if two nodes ask for the same coordinate to split, only one request will succeed, while the other one will be canceled.

Coordinate Selection. At coordinate setup, if there are more neighbors which already participate the W-Grid network, the joining sensor must choose one of them from which to take a coordinate. The selection strategy we adopt is to choose the shortest coordinate² in terms of number of bits. If two or more strings have the same length the sensor randomly chooses one of them. Experiments have shown that this policy of coordinate selection reduces as much as possible the average coordinates length in the system. In Figure 1 there is a small example of a W-Grid network. In the tree structure, parent-child relationships can be set only by nodes that are capable of bi-directional direct communication.

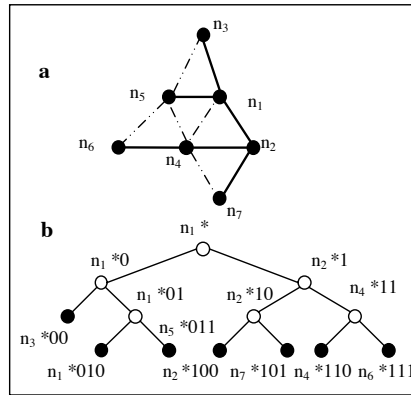


Fig. 1. Physical (a) and logical (b) network. Empty circles represent split coordinates, full black circles are coordinates that can still be split.

3.2 Formal Model: Network Properties

The sensor network is represented as a graph S :

$$S = (D, L)$$

² Among the ones that still can be split, see Coordinate Setup.

in which D is the set of participating devices and L is the set of physical connectivity between couples of devices:

$$L = \{(d_i, d_j) : \text{two-way connection between } d_i \text{ and } d_j\}$$

Each device is assigned one or more (virtual) coordinate(s). We define C as the set of existing coordinates. Each coordinate c_i is represented as a string of bits starting with \star . According to the regular expression formalism coordinates are defined as follows:

$$C = \{c : c = \star(0 | 1)^*\}$$

E.g. $\star 01001$ is a valid W-Grid coordinate. Given a coordinate c_i and a bit b their concatenation will be indicated as $c_i b$. E.g. considering $c_i = \star 0100, b = 0$ then $c_i b = \star 01000$. Given a bit b its complementary \bar{b} is defined. E.g. $\bar{1} = 0$. Some functions are defined on C :

$$\text{length}(c) : C \rightarrow \mathbb{N} \quad (1)$$

Given a coordinate c , $\text{length}(c)$ returns the number of bits in c . (\star excluded). E.g. $\text{length}(\star 01001) = 5$.

$$\text{bit}(c, k) : (C, \mathbb{N} - \{0\}) \rightarrow \{0, 1\} \quad (2)$$

Given a coordinate c and a positive integer $k \leq \text{length}(c)$, $\text{bit}(c, k)$ returns the k -th bit of c . Position 0 is out of the domain since it is occupied by \star .

$$\text{pref}(c, k) : (C, \mathbb{N}) \rightarrow C \quad (3)$$

Given a coordinate c and a positive integer $k \leq \text{length}(c)$, $\text{pref}(c, k)$ returns the first k bits of c . E.g. $\text{pref}(\star 01001, 3) = \star 010$. We define the complementary (buddy) of a coordinate c as:

$$\bar{c} = \text{pref}(c, \text{length}(c) - 1) \overline{\text{bit}(c, \text{length}(c))} \quad (4)$$

E.g. $\overline{\star 01001} = \star 01000$.

$$\text{father}(c) : (C - \{\star\}) \rightarrow C$$

$$\text{father}(c) = \text{pref}(c, \text{length}(c) - 1) \quad (5)$$

$$lChild(c), rChild(c) : (C) \rightarrow C$$

$$lChild(c) = c0 \quad (6)$$

$$rChild(c) = c1 \quad (7)$$

E.g. Given a coordinate $c_i = \star 011$, $father(\star 011) = \star 01$, $rChild(\star 011) = \star 0111$, $lChild(\star 011) = \star 0110$. A function M maps each coordinate c to the device holding it:

$$M : C \rightarrow D$$

A W-Grid network is represented as a graph:

$$W = (C, P)$$

P is the set of *parentships* between coordinates.

$$P = \{(c_i, c_j) : c_j = c_i(0 \mid 1)\}$$

E.g. $p_i = (\star 010, \star 0101)$. We define the complementary (buddy) of a parentship $p = (c_i, c_j)$ as:

$$\bar{p} = (c_i, \bar{c}_j) \tag{8}$$

E.g. $p = (\star 010, \star 0101)$, $\bar{p} = (\star 010, \star 0100)$. A graph W is a valid W-Grid network if both the following properties are satisfied:

1. $\forall p = (c_i, c_j) \in P, (M(c_i) = M(c_j)) \vee ((M(c_i), M(c_j)) \in L)$
2. $\forall p = (c_i, c_j) \in P : M(c_i) \neq M(c_j) \Rightarrow \exists \bar{p} = (c_i, \bar{c}_j) \in P : M(c_i) = M(\bar{c}_j)$

3.3 Formal Model: Network Generation

W-Grid network is generated according to this few simple rules:

1. The first node that joins the networks (that initiate a coordinate space) gets the coordinate \star . A node that holds a W-Grid coordinate is marked as **active**. A function *last* is defined:

$$last(d) : D \rightarrow C$$

which returns the last coordinate received by d . If d is **not active** the function returns $\{\emptyset\}$. After the first node, let us say n_1 , has joined the network, $last(n_1) = \star$.

2. $\forall l = (d_i, d_j) \in L : last(d_i) \neq \{\emptyset\}$ two parentships are generated:

- $p = (last(d_i), c') : M(c') = d_j$
- \bar{p}

Where $c' = lChild(last(d_i)) \mid rChild(last(d_i))$. Namely c' corresponds to the non-deterministic choice of one of the children of c . Nodes progressively get new coordinates from their physical neighbors in order to establish parentships with them. The number of coordinates at nodes may vary, in W-Grid that measure is always used as a parameter. The policies for coordinates may be: (1) a fixed number of coordinates per node (e.g. a given k) or (2) one coordinate per physical neighbor. Coordinates getting is also called split. The actors of the split procedure are an asking node and a giving node. A coordinate c_i is split by concatenating a bit to it and then, one of the new coordinates is assigned to the joining node, while the other one is kept by the giving node. Obviously, an

already split coordinate c_i can not be split anymore since this would generate duplicates. Besides, in order to guarantee coordinates' univocity even in case of simultaneous requests, each asking node must be acknowledged by the giving node. Thus, if two nodes ask for the same coordinate to split, only one request will succeed, while the other one will be temporarily rejected and postponed. Coordinate discovering is gradually performed by implicit overhearing of neighbor sensors transmissions.

3.4 Routing Algorithm

W-Grid maps nodes on an indexing binary tree T in order to build a totally ordered set over them. Each node of the tree is assigned a W-Grid virtual coordinate (c) which is represented by a binary string and has a value $v(c)$:

$$\forall c \in T, v(c) \in C$$

where C is a totally ordered set since:

$$\forall c_1, c_2 \in T : c_2 \in l(c_1) \rightarrow v(c_2) < v(c_1)$$

$$\forall c_1, c_2 \in T : c_2 \in r(c_1) \rightarrow v(c_2) > v(c_1)$$

where $r(c)$ and $l(c)$ represents the right sub-tree and the left sub-tree of a coordinate $c \in T$ respectively. And:

$$\forall c_1, c_2 \in T : F(c_1, c_2) = 0 \rightarrow v(c_1) < v(c_2)$$

$$\forall c_1, c_2 \in T : F(c_1, c_2) = 1 \rightarrow v(c_1) > v(c_2)$$

where $F(c_1, c_2)$ is a function that returns the bit of coordinate c_1 at position $i + 1$ where i corresponds to the length of the common prefix between c_1 and c_2 . For instance given two coordinates $c_1 = \mathbf{110100}$ and $c_2 = \mathbf{1110}$, $F(c_1, c_2) = 0$ therefore $c_2 > c_1$. As we stated before, the coordinate creation algorithm of W-Grid generates an order among the nodes and its structure is represented by a binary tree. The main benefit of such organization is that messages can always be delivered to any destination coordinate, in the worst case by traveling across the network by following parent-child relationship. The routing of a message is based on the concept of distance among coordinates. The distance between two coordinates c_1 and c_2 is measured in logical hops and correspond to the sum of the number of bits of c_1 and c_2 which are not part of their common prefix. For instance:

$$d(*\mathbf{0011}, *\mathbf{011}) = 5$$

Obviously it may happen that physical hops distance is less then the logical. Given a message and a target binary string c_t each node n_i forwards it to the neighbor that present the shortest distance to c_t . It is important to notice that each node needs neither global nor partial knowledge about network topology to route messages, its routing table is limited to information about its direct neighbors' coordinates. This means **scalability** with respect to network size.

³ While $F(c_2, c_1) = 1$, therefore $F(c_1, c_2) = \overline{F(c_2, c_1)}$.

4 W-Grid Data Management

W-Grid organizes nodes (i.e. sensors/devices) in a tree structure and distributes data (tuple or records with any kind of information) among them by translating the values of the record attributes into binary strings, namely into virtual coordinates that are used to locate the matching node where to store the strings, that is the data. The translation of record values into binary strings occurs by means of a linearization function mapping multidimensional data to one dimension with a good locality preserving behavior. Several linearization functions, such as Z curve, Hilbert curve etc., have been successfully adopted in the past for multi-dimensional data structures (see [2] for a survey) and in particular we adopted a modified version of the one proposed in [12].

Since W-Grid c_i are binary strings, we can see from Figure 2 that they correspond to leaf nodes of a binary tree. Therefore a W-Grid network acts directly as a distributed database with a distributed index. This means that each coordinate represent a portion (i.e. region) of the global data space as depicted in Figure 2. The mechanism described in subsection 3.3 and in 3.1, which generates new coordinates, corresponds to a split method that creates also new regions. Basically, from the viewpoint of data management, this split method divides the region in two half of equal volumes along a space dimension. The dimension is chosen following a simple rule: if a region r has been achieved by splitting his father region along the i -th dimension, then r will be split on the successive dimension, namely i -th+1 modulo number-of-space-dimensions.

An additional concept related to region splits, which is specific of the data management feature, is that all region have a maximum bucket size b that fixes the maximum number of data managed by each region. When the number of any

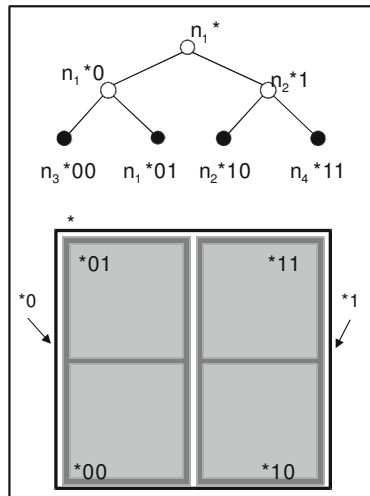


Fig. 2. Correspondence between coordinates and data space partitions

region data is equal to $b + 1$ (region overflow) then the region undergoes a split following the same method just described, but with a slight difference: if after the split one of the resulting region is still in overflow, then the split process continues recursively and stops when no region is overflowing its bucket. The process converges quickly because the region is always divided in two half and moreover it is sufficient to separate only one (overflowing) region data from the other.

The region bucket size allows also a first indirect balancing of the storage load of regions at nodes, moreover each nodes may receive several coordinates/regions. Coordinates that have been split (the empty circles in Figures 1 and 2) cannot contain data.

Let us describe a brief example of an environment monitoring application in which sensors survey temperature (T) and pressure (P), to which we refer as d_1 and d_2 . Each event is inserted in the distributed database implicitly generated by W-Grid, reporting for instance date and time of occurrence.

Without loss of generality we can define a domain for T and P let us say $Dom(d_1) = [-40, 60]$ and $Dom(d_2) = [700, 1100]$. We present an example of range query submitted to the network. *Return the events having a temperature ranging from 26 to 30 Celsius degrees and pressure ranging from 1013 to 1025mbar.* After calculating the correspondent binary string⁴ for the four corners of the range query, namely:

$$\begin{aligned} &(26,1013) \quad (26,1025) \quad (30,1013) \quad (30,1025) \\ &c_1 = *11011000 \quad c_2 = *11011001 \\ &c_3 = *11011010 \quad c_4 = *11011011 \end{aligned}$$

all we have to do is querying sensors whose coordinates have $*110110$ as prefix.

To do this we will route the range query toward $*110110$. Once the correspondent sensor has been reached it will be in charge to (1) solve part of the query if it is managing regions covered by the range query and (2) find out which of its child nodes (neighbor nodes) has coordinates that are covered by the range query. The query is then forwarded to each of these child node for further solving. We have fully implemented this algorithm and its performances are reported in Section 6.

5 Local Learning

This method introduces a learning algorithm locally at sensors, with no extra cost in terms of radio transmissions, whose goal is to learn information regarding direct sensor neighbors. This strategy improves routing performances by reducing the number of hops, namely the number of forwards, and consequently the

⁴ For instance, by standardizing 26 and 1013 (c_1) to their domains we obtain 0, 66 and 0, 783 respectively. We multiply both of them by 2^4 in order to get a string of length 8. The binary conversion of the multiplications are **1010** and **1100** respectively. Then, by crossing bit by bit the two string we get the c where destination node location is stored ***11011000**.

routing latency. The basic idea is to exploit the implicit overhearing that radio communications cause. In fact every time that a packet p_i (data or query) with destination c_i is forwarded by a sensor d_f (forwarder) to a sensor d_r (receiver), each sensor that is within the radio range (neighbors N) of d_f is aware that the packet is being forwarded. As a consequence each sensor in N can discover (i) which virtual coordinate is the destination of p_i , (ii) which sensor d_r has been chosen towards that destination and (iii) which is the distance of p_i at d_r . Here comes the local learning. If any sensor in N , let us say d_l finds out that a neighbors, let us say d_{nf} with coordinate c_{nf} would have taken p_i closer than d_r then d_l temporarily stores the pair (d_{nf}, c_i) so that when it performs the next beaconing it informs d_f that a better path has been discovered. In this way, the next time that d_f needs to forward a packet to a destination whose prefix is c_i , d_l will be preferred to d_r . Figure 3 shows an example of local learning. Packet p_i with destination $*011$ must be routed by node d_f . By forwarding p_i to d_l the distance from d_f to the destination is 3 while by forwarding p_i to d_r the distance is 5. Local learning allows n_f to know that d_l is a better choice for routing packets whose destination is $*011$ %⁵.

A possible variation of the strategy is to choose between d_l and d_r according to a certain probability, so that possible changes in network topology and consequently new possible paths can be caught even if some learning has already occurred.

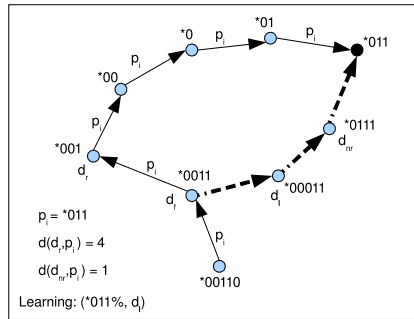


Fig. 3. Example of local learning at node n_f

6 Experimental Results

We have compared the performances of W-Grid algorithm with DIM [6] by implementing DIM in our Java Network Simulator.

We simulated four kinds of network deployment on an area of 800×800 meters in which 205 sensors were spread according to (1) uniform and (2) not uniform distribution and in both cases we generated two sets of data based (a) on a random and (b) on a skewed distribution respectively. We varied nodes

⁵ % means a binary string of arbitrary length.

densities by adjusting nodes transmission range (73, 101 and 122 meters) so that each sensor could have, on average, 4, 8 and 12 neighbors respectively. Sensors performed periodic beaconing so that coordinate creation was gradual, the simulation randomly chose one sensor to beacon first and elect itself as root of a virtual coordinate space. Then, as described in Section 3 we let sensors build the W-Grid network and a DIM network as well. Once both W-Grid and DIM network generation were completed we performed 2000 data insertion, with data generated into domains $D_1 = \{0, 800\}$ and $D_2 = \{0, 800\}$. After that we randomly generated 5000 range queries and injected them into the network to randomly chosen sensors. When creating a range query we followed these steps:

- Generation of a query central point (x, y) on D_1 and D_2
- Generation of the range by using Math.Gaussian Java function and multiplying the resulting value by a factor 70
- Applying the range to (x, y)

By fixing the factor to 70 we obtain that about 67% of the queries will have a range within 140 and 99% of them will have a range within 420. From simulations results we obtained that the 5000 range queries looked for 100000 data on average, meaning that each query covered an average of 20 data.

6.1 Network Traffic Comparison

When comparing DIM and W-Grid it is appropriate to make some considerations. DIM relies on GPSR when performing routing, this means that sensors need to be aware both of their physical location and the network perimeter. These constraints increase the cost of each sensor and limit the DIM usage possibility, for instance it cannot be used in indoor environments and in outdoor

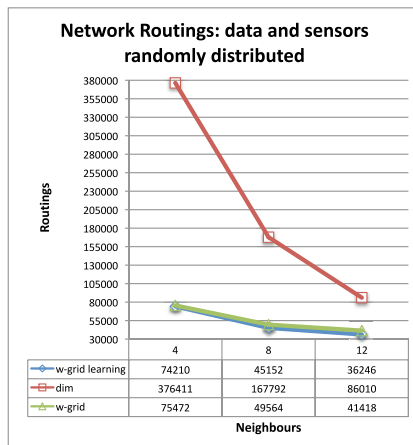


Fig. 4. Number of routings with sensors and data randomly distributed

areas where the density of sensors is beyond the GPS precision, or when weather conditions are bad.

W-Grid achieves significantly better routing performances than DIM in all of the four scenarios achieved by combining the two distributions of sensors with the two distributions of data. Moreover, the local learning improves the performance in all experiments achieving the best gain of 12% when not random data are distributed over randomly positioned sensors (see Figure 5). In all scenarios DIM reduces the wide gap to W-Grid as the network density increases. As depicted

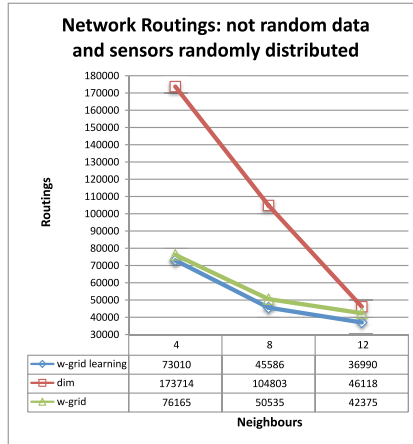


Fig. 5. Number of routings with sensors randomly distributed and data not randomly distributed

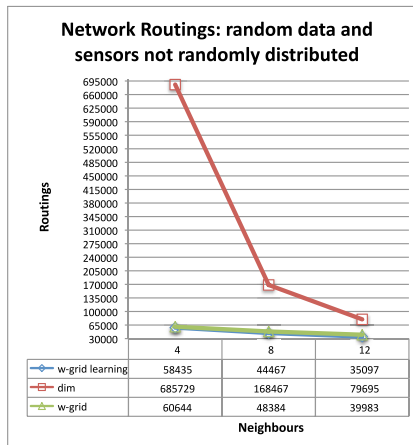


Fig. 6. Number of routings with sensors not randomly distributed and data randomly distributed

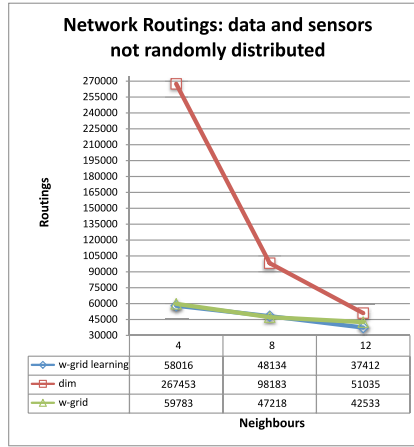


Fig. 7. Number of network routings with sensors and data not randomly distributed

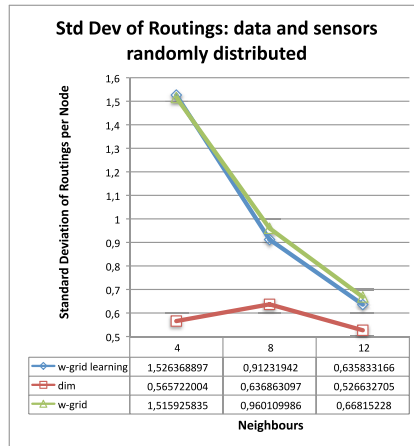


Fig. 8. Std Dev of routings over Avg routings per node, with sensors and data randomly distributed

in Figure 5 and in Figure 6, when the sensor density is 4 neighbors per sensor, DIM requires, respectively, between 3 and 10 times more routings (i.e. message forwards) than W-Grid in order to resolve the same sets of range queries over the same sensor deployments.

As far as the distribution of the routing workload per node is concerned, it is measured as the ratio between the standard deviation of the number of routings and the average of routings. When the standard deviation is greater than its corresponding average, the ratio is greater than 1 and of course it is smaller than 1 in the opposite case. It is necessary to adopt such a ratio to compare the W-Grid and DIM routing workload because the two approaches generate a

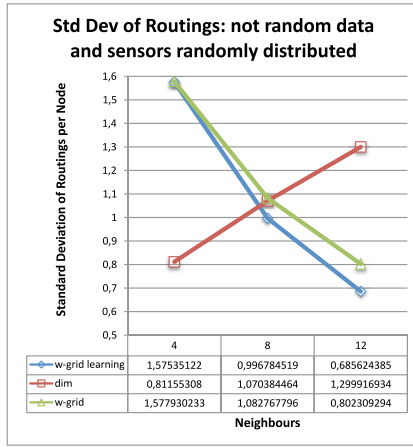


Fig. 9. Std Dev of routings over Avg routings per node, with sensors randomly distributed and data not randomly distributed

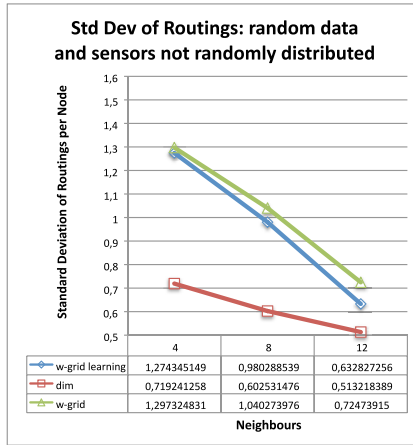


Fig. 10. Std Dev of routings over Avg routings per node, with sensors not randomly distributed and data randomly distributed

different number of routings for the same simulation configurations. As depicted in Figure 8 and 10 DIM behaves better than W-Grid when data are random. If data are not uniformly distributed, as it usually happens in real applications, W-Grid achieves a better workload balancing when the density is equal or greater than 8 neighbours per node (see Figure 9 and 11. Moreover the learning method always improves the routing workload balancing.

With regard to range queries efficacy we can observe in Figure 12 that a percentage of data between 2% and 3% are not caught by DIM range queries, while W-Grid does not miss any data. DIM losses are due to sensor placement

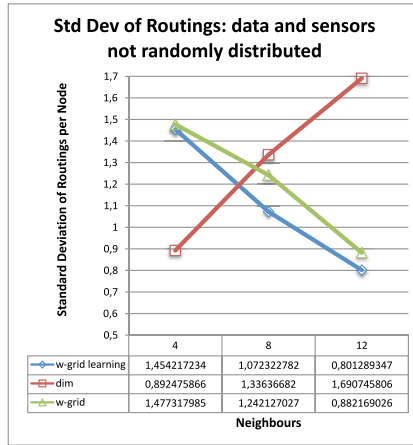


Fig. 11. Std Dev of routings over Avg routings per node, with sensors and data not randomly distributed

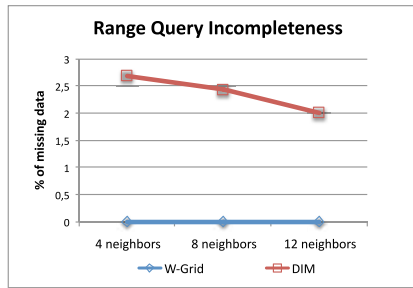


Fig. 12. Number of data not found by range queries

which may cause some regions not to be managed by any sensor and GPSR routing not being able to find the correct backup zone.

7 Conclusions

W-Grid is a cross-layering infrastructure able to self-organize wireless sensor networks for routing and multi-dimensional data management. Simulations have shown that W-Grid generates wireless networks, which significantly reduce the network traffic with respect to DIM networks in all the experimented scenarios. Moreover W-Grid produces a better balancing of the routing workload when data are not uniformly distributed and when the sensor density is equal or greater than 8 neighbours per sensor. Finally, the local learning method has further improved both the network traffic and the routing balancing of W-Grid in all experiments.

References

1. Bonnet, P., Gehrke, J., Seshadri, P.: Towards sensor database systems. In: Tan, K.-L., Franklin, M.J., Lui, J.C.-S. (eds.) MDM 2001. LNCS, vol. 1987, pp. 3–14. Springer, Heidelberg (2000)
2. Gaede, V., Günther, O.: Multidimensional access methods. *ACM Computing Surveys* 30(2), 170–231 (1998)
3. Greenstein, B., Estrin, D., Govindan, R., Ratnasamy, S., Shenker, S.: Difs: A distributed index for features in sensor networks. In: Proceedings of first IEEE WSNA, pp. 163–173. IEEE Computer Society, Los Alamitos (2003)
4. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.* 11(1), 2–16 (2003)
5. Karp, B., Kung, H.: GPSR: greedy perimeter stateless routing for wireless networks. In: *MobiCom 2000: 6th annual international conference on Mobile computing and networking*, pp. 243–254. ACM Press, New York (2000)
6. Li, X., Kim, Y., Govindan, R., Hong, W.: Multi-dimensional range queries in sensor networks. In: *SenSys 2003: Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 63–75. ACM Press, New York (2003)
7. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.* 36(SI), 131–146 (2002)
8. Monti, G., Moro, G.: Multidimensional Range Query and Load Balancing in Wireless Ad Hoc and Sensor Networks. In: Proceedings of the Eighth IEEE International Conference on Peer-to-Peer Computing (P2P 2008), pp. 205–214 (2008)
9. Monti, G., Moro, G.: Scalable multi-dimensional range queries and routing in data-centric sensor networks. In: *Infoscale 2008: The Third International ICST Conference on Scalable Information Systems* (2008)
10. Monti, G., Moro, G., Lodi, S.: W*-Grid a robust decentralized cross-layer infrastructure for routing and multi-dimensional data management in wireless ad-hoc sensor networks. In: *P2P 2007: Seventh IEEE International Conference on Peer-To-Peer Computing*, pp. 159–166 (2007)
11. Moro, G., Monti, G.: W-Grid: a self-organizing infrastructure for multi-dimensional querying and routing in wireless ad-hoc networks. In: *P2P 2006: Sixth IEEE International Conference on Peer-To-Peer Computing*, pp. 210–220 (2006)
12. Ouksel, M.A.: The interpolation based grid file. In: *ACM SIGACT-SIGMOD 1985: Proceedings of Symposium on Principle of Database Systems*, pp. 20–27. ACM Press, New York (1985)
13. Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., Yin, L., Yu, F.: Data-centric storage in sensornets with ght, a geographic hash table. *Mob. Netw. Appl.* 8(4), 427–442 (2003)
14. Xiao, L., Ouksel, A.: Tolerance of localization imprecision in efficiently managing mobile sensor databases. In: *MobiDE 2005: Proceedings of the 4th ACM international workshop on Data engineering for wireless and mobile access*, pp. 25–32. ACM Press, New York (2005)
15. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. In: *MobiCom 2002: Proc. of the 8th annual international conference on Mobile computing and networking*, pp. 148–159. ACM Press, New York (2002)

QShine 2009

**Invited Session VI – Performance
Optimization and Device Heterogeneity
in Wireless Networks**

Performance Analysis and Cross Layer Optimization for Multimedia Streaming over Wireless Networks

Antonio Ao, Zhung-Han Wu, and Ping-Cheng Yeh

Graduate Institute of Communication Engineering,
National Taiwan University, Taipei, Taiwan
{r96942044, f97942031, pcyeh}@ntu.edu.tw

Abstract. Multimedia data are sensitive to delay which deteriorates the video quality and the perception of the viewer. Due to the sensitivity to the delay, transmitting multimedia data over inherently variant wireless channels is a big challenge. Although wireless systems have employed adaptive modulation and coding (AMC) to combat the variation of the environment, the possible delay in the buffers still has impacts on the video quality. In this paper, the focus is on the performance of wireless multimedia streaming using AMC. In particular, the video frame error rate and the resulting GOP distortion for video streaming is analyzed. It is observed that the video quality depends on the target packet error rate of the AMC, which also determines the SNR thresholds of AMC. Through our analysis, the optimal target PER of AMC and the resulting SNR thresholds can be obtained to achieve the optimal video quality that minimizes the GOP distortion.

Keywords: queueing analysis, cross layer optimization, multimedia, streaming, AMC, distortion.

1 Introduction

Multimedia streaming faces a big challenge in delivering multimedia data within strict delay bound for the receiving node to reconstruct the video streaming in time. On the other hand, the nature of the varying wireless channels results in even bigger challenge for wireless multimedia streaming. Different techniques have been proposed to deal with the varying channels. Adaptive modulation and coding (AMC) is a technique commonly used which dynamically adjusts the modulation and coding according to the channel condition to maximize the overall throughput. When multimedia streaming application is performed upon wireless networks with AMC, the video quality is significantly affected by AMC. When the channel condition is bad, AMC transmits data with low rate mode to ensure low probability of error during the transmission. Yet, there is a tradeoff here. Due to the low transmission rate of AMC under the poor channel state, the data packets may be queued in the buffer and experience longer delays due to bad channel conditions. Under such circumstances, the delay sensitive multimedia packets may be dropped due to timeouts and buffer overflows. The dropped packets thus cause severe deterioration to the quality of the reconstructed video.

In the literature, several works explore the effect of AMC on the queueing behavior of wireless transmissions. The first study analyzing the queueing behavior of wireless

transmissions with AMC is given in [1]. The queueing analysis is extended for AMC with automatic repeat request (ARQ) incorporated [2]. In [3], the queueing analysis with AMC over MIMO system is studied. It is noted that the works in [1, 2, 3] do not consider the timeout problem of delay-sensitive traffic when transmitting with AMC. There have been some works focusing on the issue. In [4], the authors consider the problem of scalable video transmission with adaptive BCH codes. However, BCH codes are not commonly used in practical AMC. In [5, 6], the timeout probability of video traffic over AMC transmission is analyzed using the effective capacity method. The quality of the reconstructed video and the effect of cross traffic from other multimedia streams are not considered in these works.

In this paper, a cross-layer optimization algorithm for maximizing video quality is proposed. The queue of multimedia streaming via AMC is first analyzed. The analysis is then applied to analyze the video frame error rate (VFER) and the resulting GOP distortion for the video streaming. The effect of the interruption of cross traffic from other multimedia streams is also analyzed. It is observed that the video quality depends on the target packet error rate (PER) of AMC, which also determines the SNR thresholds of AMC. Through our analysis, the optimal target PER of AMC and the resulting SNR thresholds can be obtained to achieve the optimal video quality that minimizes the GOP distortion.

The remainder of the paper is organized as follows. In Section 2, the system model is first described. In Section 3, we analyze the video streaming quality for the case of single multimedia stream. In Section 4, we extend the work to the case of multiple streams. The analysis in these sections enables us to determine the optimal AMC for video streaming. In Section 5, the numerical results are presented. Finally, the conclusions are given in Section 6.

2 System Model

2.1 System Description

The block diagram of wireless multimedia transmission from a base station to a mobile receiver through fading channels is illustrated in Figure 1. At the base station, a classifier divides the arriving packet into two categories, delay sensitive multimedia packets and delay insensitive data packets. The two classes of packets are sent to two different buffers, the media buffer and the data buffer respectively. In this work, we consider a general model which allows the base station to serve multiple multimedia streams at the same time. At the mobile receiver, it estimates the channel quality to decide the AMC mode to use. The AMC mode choice is then fed back to base station for the PHY mode controller to adjust the modulation/coding for packet transmissions.

2.2 Packet/MAC Frame Structure

Real-time multimedia packet transmissions employ UDP protocol to avoid the extra delay caused by retransmissions. As shown in Figure 2, a UDP packet contains header, payload and CRC. The packet accommodates $N_b = 11680$ bits (1460 bytes). The CRC

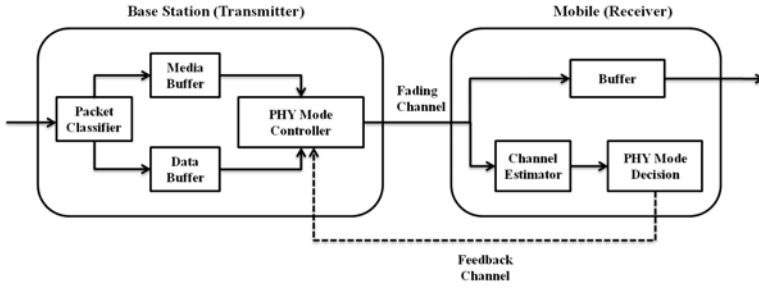


Fig. 1. Wireless multimedia transmissions via AMC

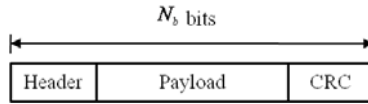


Fig. 2. Packet structure

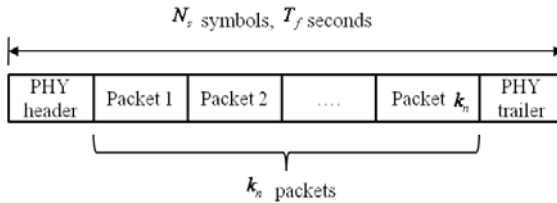


Fig. 3. MAC frame structure

is assumed perfect for packet error detection. If a packet is detected to be in error, the packet is dropped and declared packet loss. The header and the CRC parity bits are assumed to be negligible for throughput calculation.

Fig. 3 illustrates the MAC frame structure. Each MAC frame consists of header, trailer and packets coming from upper layer. Each MAC frame contains N_s symbols and the MAC frame duration is T_f seconds. Given the AMC mode is in mode n with transmission rate R_n , the value of N_s can be calculated by

$$N_s = N_{overhead} + \frac{k_n N_b}{R_n} \approx \frac{k_n N_b}{R_n}, \quad (1)$$

where k_n is the number of packets in a MAC frame, and $N_{overhead}$ denotes the number of symbols of physical layer overhead. Note that $N_{overhead}$ can be ignored because it is much less relative to the number of payload symbols. Thus

$$k_n \approx \frac{N_s}{N_b} R_n = b R_n \quad (2)$$

where $b \triangleq N_s/N_b$. b is the number of total packets grouped together per MAC frame given rate R_n .

2.3 AMC Transmission Modes and Packet Error Rate

In this paper, the following transmission mode (TM) for physical layer AMC is considered. The transmission modes use M_n QAM modulation with punctured convolutional codes. Monte Carlo simulation is conducted to obtain the exact PER. The simulation block diagram is depicted in Figure 4. The generator polynomial of mother code is $g = [133, 171]$, and the coding rate and puncturing pattern are adopted from IEEE 802.11a [7]. The packet size is set to be the same as N_b (1460 bytes). For the sake of analysis, the simulated PER of each AMC mode n is fitted by a piecewise function $p_n(\gamma)$ as

$$p_n(\gamma) = \begin{cases} 1, & \text{if } \gamma < \gamma_{pn} \\ \exp(a_n - g_n\gamma), & \text{if } \gamma \geq \gamma_{pn}, \end{cases} \quad (3)$$

Simulation and fitting results are shown in Figure 5. The figure shows the fitting curves well match the simulation curves. In Table II we summarize the fitting results of each transmission mode.

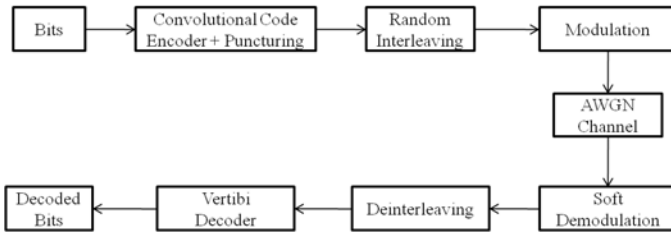


Fig. 4. Block diagram of TM simulation

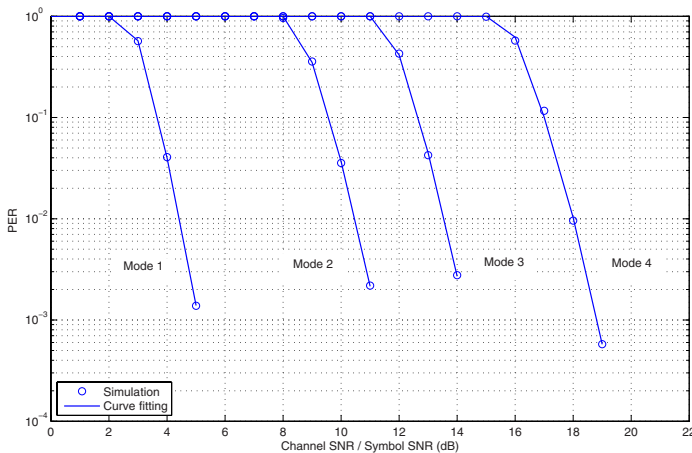


Fig. 5. Packet error rate versus channel SNR in different modes, packet size = 1460 bytes

Table 1. Curve fitting parameters in different modes

	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
Modulation	-	QPSK	16-QAM	16-QAM	64-QAM
Coding Rate R_c	-	1/2	1/2	3/4	2/3
R_n (bits/sym.)	0	1	2	3	4
a_n	-	9.7413	7.6572	7.7266	6.5429
g_n	-	5.1606	1.0960	0.5430	0.1022
γ_{pn} (dB)	-	3	9	12	16

2.4 FSMC Wireless Channel Model

FSMC is widely used to model the varying wireless channel which matches well with the actual channel measurement experiments. The channel SNR is partitioned into $N+1$ non-overlapping consecutive intervals by the SNR threshold points $\{\gamma_n\}_{n=0}^{N+1}$, and each SNR interval is associated with an AMC mode. When the received channel SNR γ falls in $[\gamma_n, \gamma_{n+1})$, the channel is said to be in channel state n and AMC mode n is used for transmission. The SNR thresholds $\{\gamma_n\}_{n=0}^{N+1}$ are determined by the QoS criteria in the following section. Based on [9] and [10], the probability of the channel being in state n is

$$\begin{aligned} \Pr(n) &= \int_{\gamma_n}^{\gamma_{n+1}} p(\gamma) d\gamma \\ &= \frac{\Gamma(m, \frac{m\gamma_n}{\bar{\gamma}}) - \Gamma(m, \frac{m\gamma_{n+1}}{\bar{\gamma}})}{\Gamma(m)}, \end{aligned} \quad (4)$$

where

$$\Gamma(m, x) = \int_x^{\infty} t^{m-1} \exp(-t) dt$$

is the complementary incomplete Gamma function, and

$$p(\gamma) = \frac{m^m \gamma^{m-1}}{\bar{\gamma}^m \Gamma(m)} \exp(-\frac{m\gamma}{\bar{\gamma}}) \quad (5)$$

is the probability density function (pdf) of the commonly used Nakagami- m distribution for modeling the receive channel SNR. Note that $\bar{\gamma} = E\{\gamma\}$ denotes the average received SNR and $\Gamma(m) = \int_0^{\infty} t^{m-1} \exp(-t) dt$ is the Gamma function.

Since the channel is varying continuously, it is generally assumed that the channel state transition probability $P_{m,n} = 0$ for any nonconsecutive states m, n such that $|m - n| \geq 2$. The remaining nonzero transition probabilities are the adjacent-state transition probabilities which can be computed as [11]

$$\begin{aligned} P_{n,n+1} &= \frac{N_{n+1} T_f}{\Pr(n)}, \text{ if } n = 0, 1, \dots, N-1. \\ P_{n,n-1} &= \frac{N_n T_f}{\Pr(n)}, \text{ if } n = 1, \dots, N, \end{aligned} \quad (6)$$

where N_n denotes the level crossing rate of state n . The value of N_n can be approximated by [12]

$$N_n = \frac{\sqrt{2\pi}f_d}{\Gamma(m)} \left(\frac{m\gamma_n}{\bar{\gamma}}\right)^{m-0.5} \exp\left(-\frac{m\gamma_n}{\bar{\gamma}}\right), \tag{7}$$

where f_d denotes the maximum Doppler shift. The $(N + 1) \times (N + 1)$ channel state transition probability matrix P_c can then be expressed as

$$\mathbf{P}_c = \begin{bmatrix} 1 - P_{0,1} & P_{0,1} & \cdots & 0 \\ P_{1,0} & 1 - P_{1,0} - P_{1,2} & P_{1,2} & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & P_{N-1,N-2} & 1 - P_{N-1,N-2} - P_{N-1,N} & P_{N-1,N} \\ 0 & \cdots & P_{N,N-1} & 1 - P_{N,N-1} \end{bmatrix}. \tag{8}$$

2.5 QoS Criteria and AMC SNR Thresholds

In this paper, we consider the QoS criteria which requires the target average PER of the AMC, $\overline{\text{PER}}$, to be P_0 . The average PER of AMC in mode n is of the form [10]

$$\begin{aligned} \overline{\text{PER}}_n &= \frac{1}{\text{Pr}(n)} \int_{\gamma_n}^{\gamma_{n+1}} \exp(a_n - g_n\gamma) p(\gamma) d\gamma \\ &= \frac{1}{\text{Pr}(n)} \frac{\exp(a_n)}{\Gamma(m)} \left(\frac{m}{\bar{\gamma}}\right)^m \frac{\Gamma(m, b_n\gamma_n) - \Gamma(m, b_n\gamma_{n+1})}{(b_n)^m} \\ &, n = 1, \dots, N, \end{aligned} \tag{9}$$

where

$$b_n = \frac{m}{\bar{\gamma}} + g_n. \tag{10}$$

The average PER of AMC can then be obtained as the ratio of the average number of error packets over the total average number of transmitted packets. Together with the QoS criteria, we have the following equation

$$\overline{\text{PER}} = \frac{\sum_{n=1}^N R_n \text{Pr}(n) \overline{\text{PER}}_n}{\sum_{n=1}^N R_n \text{Pr}(n)} = P_0. \tag{11}$$

One can find the AMC SNR thresholds $\{\gamma_n\}_{n=0}^{N+1}$ from the equation above following the algorithm in [11], which we omit the details here.

2.6 Queuing Model

Detailed description of the queuing model is given as follows.

Queuing Policy. The media buffer and the data buffer are both of finite buffer lengths. The time is divided into slots, each of length equal to MAC frame duration T_f . Each

buffer operates following the first-come first-serve (FCFS) policy. When the buffer is full, the newly arrived packets are dropped. Each packet in the queue waits for at most P time slots for service before it gets timeout and discarded. There is no retransmission when multimedia packet transmission error occurs so as not to introduce extra delay. Throughout the paper, it is assumed that the media buffer has higher priority over the data buffer. The data packets are served only when the media server is empty. Thus the focus is on analyzing the performance of the media queue in this paper.

Arrival Process. For the media queue, it is assumed that the aggregated packet arrival from multiple multimedia streams is a Poisson process with mean λT_f . The probability mass function of the number of arrival is

$$P(A = k) = \begin{cases} \frac{(\lambda T_f)^k \exp(-\lambda T_f)}{k!}, & \text{if } k \geq 0 \\ 0, & \text{else,} \end{cases} \quad (12)$$

where $A \in \{0, 1, 2, \dots\}$. In the case of multiple multimedia streams, we assume the video packets of each multimedia stream arrive in a batch during each slot due to the packet aggregation of the stream in the previous hop. This indicates that in each slot, packets from certain stream A either all arrive before or all arrive after the packet batch of another stream B. It is assumed that there are no interlacing of packets from two different multimedia streams within each slot.

Service Policy. The service rate of the queue dynamically adjusts according to the channel state and the associated AMC mode. When the FSMC state is n , the AMC is in mode n which transmits c_n packets per time slot. From (2), it is noted that $c_n = k_n = bR_n$. The set of all possible service rates is denoted by $\Psi = \{c_0, c_1, \dots, c_N\}$. At the beginning of the slot, the server discards the time-out packets which have waited for P time-units. The server sends packets out of the queue at the beginning of the slot based on the service rate $c \in \Psi$ of the slot. It is also noted that the packet arrivals within each slot can not be served until the next slot.

3 Performance Analysis for Single Stream Case

In this Section, the queue is studied for the case of single multimedia stream. In particular, we induce a refined Markov chain from the queue. The state of the refined Markov chain specifies the current service rate resulted from AMC, and the delays experienced by the current packets in the queue. The transition probabilities of the refined Markov chain is derived and it enables us to find the steady state probability of the refined Markov chain. The steady state probability is further used for analyzing the video quality of the multimedia stream.

3.1 Refined Markov Chain

Let $S_{(c, u_1, \dots, u_P)}$ denote the state of the refined Markov chain induced from the media queue, where $c \in \Psi$ denotes the service rate of the server at the time and u_i denotes

the number of packets having waited i time slots in the queue. In particular, u_P denotes the number of packets having waited P time slots and thus exceeded their lifetimes. Since the media buffer is of finite length K , we can express the space of all valid $\underline{u} = (u_1, \dots, u_P)$ as

$$\underline{u} \in \Omega, \quad \Omega = \left\{ (u_1, u_2, \dots, u_P) \left| \begin{array}{l} u_i \in \{0, 1, \dots, K\}, i = 1, \dots, P \\ \text{and } u_1 + u_2 + \dots + u_P \leq K \end{array} \right. \right\}. \quad (13)$$

For the sake of clarity, we also use the notation of $S_{(c, \underline{u})}$ to denote $S_{(c, u_1, \dots, u_P)}$ throughout our work. The total number of states for the refined Markov chain can be is

$$L = (N + 1) \sum_{n=0}^K H_n^P = (N + 1) \sum_{n=0}^K \frac{(P + n - 1)!}{(P - 1)!n!}. \quad (14)$$

Based on the queueing model in Section 2.6, one can observe that for the refined Markov chain of current slot to transit from $S_{(c, \underline{u})} = S_{(c, u_1, \dots, u_P)}$ to $S_{(d, \underline{v})} = S_{(d, v_1, \dots, v_P)}$ in the next slot, where (c, \underline{u}) and (d, \underline{v}) must satisfy the the following constraints:

1. For the element v_P ,

$$v_P = \max\{0, u_{P-1} - c\}. \quad (15)$$

2. For the v_{P-n} term, where $1 \leq n < P - 1$,

$$v_{P-n} = \begin{cases} \max\{0, \sum_{j=1}^{n+1} v_{P-j} - c\}, & \text{if } v_{P-n+1} = 0 \\ u_{P-n-1}, & \text{if } v_{P-n+1} \neq 0. \end{cases} \quad (16)$$

3. Given the current state of the refine Markov chain is $S_{(c, \underline{u})}$, the number of packets remaining in the queue within after the packet transmissions of the current slot is

$$L_{(c, \underline{u})} = \max\{0, \sum_{j=1}^{P-1} u_j - c\}, \quad (17)$$

and the number of free space in the queue is

$$F_{(c, \underline{u})} = K - L_{(c, \underline{u})} = K - \max\{0, \sum_{j=1}^{P-1} u_j - c\}. \quad (18)$$

If there are total of A packets arriving during the slot, it is obvious to see that v_1 satisfies

$$v_1 = \begin{cases} A, & \text{if } A < F_{(c, \underline{u})} \\ F_{(c, \underline{u})}, & \text{if } A \geq F_{(c, \underline{u})}. \end{cases} \quad (19)$$

From the constraints above, we can derive the $L \times L$ transition probability matrix of the refined Markov chain. Define the transition probability matrix as

$$\mathbf{P} = [P_{(c, \underline{u}), (d, \underline{v})}], \quad (20)$$

where $P_{(c, \underline{u}), (d, \underline{v})}$ denotes the transition probability from $S_{(c, \underline{u})}$ to $S_{(d, \underline{v})}$, $c, d \in \Psi$ and $\underline{u}, \underline{v} \in \Omega$. The transition probability depends on the service rate transition, arrival

process, and waiting times experience by the current packets in the queue. It can be derived as

$$P_{(c,d),(\underline{u},\underline{v})} = P(S_{(d,\underline{v})}|S_{(c,\underline{u})}) = P(d|c)P(\underline{v}|S_{(c,\underline{u})}) = P_{c,d}P(\underline{v}|S_{(c,\underline{u})}), \quad (21)$$

where $P_{c,d}$ is the transition probability of the service rate. Note that the service rate transition is determined by the FSMC state transition. If $c = c_m$ and $d = d_n$, $P_{c,d} = P_{c_m,d_n} = P_{mn}$ of matrix \mathbf{P}_c in (8). On the other hand, for $P(\underline{v}|S_{(c,\underline{u})})$, it is easy to see $P(\underline{v}|S_{(c,\underline{u})}) = 0$ if $c, \underline{u}, \underline{v}$ do not satisfy the constraints (15), (16). Thus we have

$$P(\underline{v}|S_{(c,\underline{u})}) = \begin{cases} P(v_1|S_{(c,\underline{u})}), & \text{if } \underline{v}, c \text{ and } \underline{u} \text{ satisfy (15), (16)} \\ 0, & \text{else,} \end{cases} \quad (22)$$

with

$$P(v_1|S_{(c,\underline{u})}) = \begin{cases} P(A = v_1), & v_1 < F_{(c,\underline{u})} \\ 1 - \sum_{k=0}^{F_{(c,\underline{u})}} P(A = k), & v_1 \geq F_{(c,\underline{u})} \end{cases}. \quad (23)$$

Through (21)-(23), one can obtain \mathbf{P} in (20). Denote the steady state probability of the refined Markov chain at state $S_{(c,\underline{u})}$ by $\pi_{(c,\underline{u})}$. Since matrix \mathbf{P} is irreducible, The steady state vector $\underline{\pi} = [\pi_{(c,\underline{u})}]$ satisfies

$$\underline{\pi} = \underline{\pi}\mathbf{P}, \quad \sum_{c \in \Psi, \underline{u} \in \Omega} \pi_{(c,\underline{u})} = 1. \quad (24)$$

One can easily solve for $\underline{\pi}$ numerically.

3.2 Video Frame Error Rate Analysis

In MPEG video compression, the video sequence is divided into group of picture (GOP) for data compression. In each GOP, the video codec generates different types of frames, i.e. I-frame, P-frame and B-frame. The size of I-frame is generally larger than that of P-frame and B-frame. Due to varying length of the video frame, the VFER is different for different type of frames, which is a big challenge.

Assuming that the length of the video frame considered is fragmented into N_v packets for transmission. We assume that these N_v packets arrive the base station within one time slot. There are three possible events that causes the video frame to be in error at the receiver.

1. Blocking event \mathcal{B} : Any of the N_v packets fails to enter the media queue when the buffer is full.
2. Timeout event \mathcal{T} : Any of the N_v packets gets discarded due to waiting for P slots in the queue.
3. PHY transmission error event \mathcal{E} : Any of the N_v packets experiences transmission error when transmitted by AMC.

Define $P_{\mathcal{B}} = P(\mathcal{B})$, $P_{\mathcal{T}} = P(\mathcal{T})$, and $P_{\mathcal{E}} = P(\mathcal{E})$. These probabilities are derived as follows.

Blocking Probability. From the steady state probability $\underline{\pi}$ of the refined Markov chain, we can derive P_B as

$$P_B = \sum_{c \in \Psi, \underline{u} \in \Omega} P(\mathcal{B}|S_{(c, \underline{u})})\pi_{(c, \underline{u})}, \quad (25)$$

where $P(\mathcal{B}|S_{(c, \underline{u})})$ is the probability of frame error due to blocking packets given $S_{(c, \underline{u})}$. The probability can be expressed as

$$\begin{aligned} P(\mathcal{B}|S_{(c, \underline{u})}) &= P\{0, F_{(c, \underline{u})} - N_v < 0\} \\ &= \begin{cases} 1, & \sum_{i=1}^{P-1} u_i - c - N_v < 0 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (26)$$

Timeout Probability. The timeout probability can also be derived from $\underline{\pi}$ as

$$P_T = \sum_{c \in \Psi, \underline{u} \in \Omega'_c} P(\mathcal{T}|S_{(c, \underline{u})})\pi_{(c, \underline{u})}, \quad (27)$$

where

$$\Omega'_c = \left\{ [x_1, x_2, \dots, x_P] \mid \begin{array}{l} x_i \geq 0, x_1 + x_2 + \dots + x_P \leq K, \\ x_1 + x_2 + \dots + x_{P-1} - c \geq N_v \end{array} \right\} \quad (28)$$

is the set of possible \underline{u} which ensures that the blocking event will not occur given the current service rate c . Let d_1, d_2, \dots, d_{P-1} be the service rates in following $(P-1)$ time slots. Note that a necessary condition for any of the N_v packets to be timeout is

$$d_1 + d_2 + \dots + d_{P-1} < L_{(c, \underline{u})} + N_v = \max\{0, \sum_{j=1}^{P-1} u_j - c\} + N_v = I_{(c, \underline{u})}, \quad (29)$$

i.e. the total amount of the served packets in the next following $P-1$ time slots is less than the current queue length plus N_v . This is not a sufficient condition since it is possible to have packets in $L_{(c, \underline{u})}$ to get timeout which saves the N_v packets from time out even if the inequality does not hold. The probability of the necessary condition is an upper bound of $P(\mathcal{T}|S_{(c, \underline{u})})$. We use it as an approximation of $P(\mathcal{T}|S_{(c, \underline{u})})$, which can be computed as

$$P(\mathcal{T}|S_{(c, \underline{u})}) \lesssim \sum_{d_1 + d_2 + \dots + d_{P-1} < I_{(c, \underline{u})}} P_{c, d_1} P_{d_1, d_2} \cdots P_{d_{P-2}, d_{P-1}}, \quad (30)$$

where $\{P_{d_i, d_k}\}$ is the FSMC transition probabilities from \mathbf{P}_c in (8).

PHY Transmission Error Probability. Given that the QoS criteria demands the AMC to maintain a target PER of P_0 , P_E can simply derived as

$$P_E = \sum_{c \in \Psi, \underline{u} \in \Omega'_c} \{1 - (1 - P_0)^{N_v}\} \{1 - P(\mathcal{T}|S_{(c, \underline{u})})\} \pi_{(c, \underline{u})}. \quad (31)$$

With the three probabilities, one can see that the VFER of video frames of size N_v packets is a function of N_v and P_0 , which can be expressed as

$$\begin{aligned} \xi(N_v, P_0) &= P_B + (1 - P_B)P_T + (1 - P_B)(1 - P_T)P_E \\ &= 1 - (1 - P_B)(1 - P_T)(1 - P_E). \end{aligned} \quad (32)$$

3.3 Video Codec Performance Analysis and Optimization

The performance of a video codec is often evaluated by the GOP distortion [13] and the peak signal-to-noise ratio (PSNR). Given the target error rate P_0 of the AMC, the expected GOP distortion can be computed as

$$D_{GOP}(P_0) = D_0 - \sum_l \Delta D_l \prod_{l' \leq l} (1 - \xi_{l'}(P_0)), \tag{33}$$

where D_0 is the expected distortion if no video frame from the GOP is received, ΔD_l is the expected distortion reduction if the l -th frame of the GOP is correctly received (given the previous $(l - 1)$ frames are all correctly received). The model assumes that given any of the frame prior to the l -th frame is corrupted, the content embedded in the l -th frame is not decodable. The values of D_0 and ΔD_l are codec and content dependent. One can obtain these values simply from simulation. The VFER $\xi_l(P_0)$ is the error rate for the l -th frame in the GOP. Depending on the frame type of the l -th frame (I, P, or B) and the video content, one can find the average length N_{vl} of the l -th frame in the GOP via simulation. From the VFER analysis in the previous section, we can compute ξ_l as

$$\xi_l(P_0) = \xi(N_{vl}, P_0). \tag{34}$$

From the GOP distortion, we can compute the PSNR $\Gamma(P_0)$ as

$$\Gamma(P_0) = 10 \log_{10} \left(\frac{255^2}{D_{GOP}(P_0)} \right) \text{ (dB)}. \tag{35}$$

One can see that the GOP distortion and the PSNR is a function of P_0 . The AMC can be optimized for the video streaming performance by finding the optimal target PER P_0^* that maximizes $\Gamma(P_0)$, and then determine the SNR thresholds of the AMC from P_0^* following the algorithm in [11].

4 Performance Analysis for Multiple Streams Case

For the case of multiple multimedia streams, the cross traffic from other multimedia streams arriving the media queue earlier than the packet arrivals of the concerned stream has to be considered. In Figure 6, the arrivals from the cross traffic and the concerned

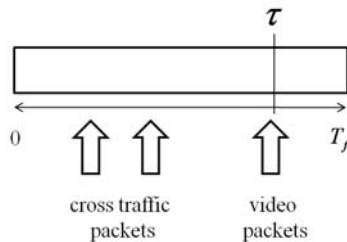


Fig. 6. Cross traffic arrival

video streams are depicted for the duration of a slot. Denote the arrival rate of the cross traffic by λ' and the arrival time of the video packet batch from the concerned stream by τ . Due to the property of Poisson arrival process, τ is of uniform distribution $U(0, T_f)$. The probability having α cross traffic packets arrival ahead of the video packets from the concerned stream can be derived as

$$\begin{aligned}
 P(A_c = \alpha) &= \int_0^{T_f} P\{\lambda'\tau = \alpha\} \cdot \frac{1}{T_f} d\tau \\
 &= \int_0^{T_f} \frac{(\lambda'\tau)^\alpha e^{-\lambda'\tau}}{\alpha!} \frac{1}{T_f} d\tau \\
 &= \frac{1}{\lambda'T_f \cdot \alpha!} \left\{ \alpha! - e^{-\lambda'T_f} \sum_{i=0}^{\alpha} \frac{\alpha!}{(\alpha-i)!} (\lambda'T_f)^{\alpha-i} \right\}. \tag{36}
 \end{aligned}$$

The queuing analysis with the presence of cross traffic is generally a difficult task. To simplify the analysis, we assume that during the steady state, the cross traffic arrivals during a slot will not cause the refined Markov chain to deviate from its steady state to a great extent. The performance of the video streaming can be analyzed simply by applying the analysis in Section 3 with modified P_B and P_T . Note that P_E is unaffected by the cross traffic.

Modified Blocking Probability. The blocking probability with the interruption of cross traffic during the steady state can be written as

$$P_B = \sum_{c \in \Psi, \underline{u} \in \Omega} P(B|S_{(c, \underline{u})}) \pi_{(c, \underline{u})}, \tag{37}$$

where

$$\begin{aligned}
 P(B|S_{(c, \underline{u})}) &= P\{A_c > \max(0, F_{(c, \underline{u})} - N_v)\} \\
 &= 1 - \sum_{\alpha=0}^{\beta_{(c, \underline{u})}} P(A_c = \alpha), \tag{38}
 \end{aligned}$$

with $\beta_{(c, \underline{u})}$ defined as

$$\beta_{(c, \underline{u})} = \max\{0, F_{(c, \underline{u})} - N_v\}. \tag{39}$$

Modified Timeout Probability. The modified timeout probability can be expressed as

$$\begin{aligned}
 P_T &= \sum_{S_{(c, \underline{u})} \in \{S_{(e, \underline{r})} | F_{(e, \underline{r})} > N_v\}} \left\{ \sum_{\alpha \leq \max\{0, F_{(c, \underline{u})} - N_v\}} P(T|S_{(c, \underline{u})}, A_c = \alpha) \right\} \\
 &\hspace{20em} P(S_{(c, \underline{u})}, A_c = \alpha) \\
 &= \sum_{c \in \Psi, \underline{u} \in \Omega'_c} \sum_{\alpha=0}^{\beta_{(c, \underline{u})}} P(T|S_{(c, \underline{u})}, A_c = \alpha) P(A_c = \alpha) \pi_{(c, \underline{u})}.
 \end{aligned}$$

Define the number of total packets in the queue including N_v video packets and cross-traffic packets as

$$I_{(c,\underline{u})} = L_{(c,\underline{u})} + N_v + A_c, \quad (40)$$

we can obtain $P(\mathcal{T}|S_{(c,\underline{u})}, A_c = \alpha)$ as

$$P(\mathcal{T}|S_{(c,\underline{u})}, A_c = \alpha) = \sum_{d_1+d_2+\dots+d_{P-1} < I_{(c,\underline{u})}} P_{c,d_1} P_{d_1,d_2} \cdots P_{d_{P-2},d_{P-1}}. \quad (41)$$

Recall that P_{d_i,d_k} are elements of \mathbf{P}_c in (8).

5 Numerical Results

In this section, the numerical experiment results are demonstrated to verify our performance analysis for the video streaming. We consider the transmission of MPEG2 video over wireless LAN. The case of two video streams streaming at the same time is simulated. Both streams have the same content and identical codec setting. The simple MPEG-2 profile is used for video compression, and the codec settings are summarized in Table 2. Note that $N_v(I)$ denotes the number of packets fragmented from I-frame and $N_v(P)$ denotes the number of packets fragmented from P-frame. The simulation parameters of the numerical experiments in this section are summarized in Table 3.

Table 2. MPEG-2 codec setting

Sequence	Foreman
Frame rate	30 frames/sec
GOP	5
Resolution	176 × 144
Chroma format	4:2:2
Bit-rate	280kbps , 380kbps
$N_v(I)$	2 packets , 3 packets respectively
$N_v(P)$	1 packet
Frame pattern	I PPPP

Figure 7 shows curves of the VFER $\xi(N_v, P_0)$ versus the target PER P_0 of the AMC. The solid lines are computed from our analysis whereas the dashed lines are obtained from simulations. One can observe that the proposed analysis is very accurate in capturing the VFER and the tradeoff associated with P_0 . When the target PER P_0 is set too small, the AMC transmits with very low rate. As a result, video packets are queued in the buffer and many of them get expired before transmission. As a result, the VFER is increased. On the other hand, when P_0 is very large, though the AMC transmits with high rate which reduces the timeout probability, the PHY errors occur with high probability. As a result, VFER is also increased. Therefore, there is an optimal choice of P_0 to minimize the VFER. The star marks in the figure denote the VFER minimum for

Table 3. Simulation Parameters

Packet size N_b	1460 bytes
Slot duration T_f	6ms
b defined in (2)	1
The mean of Poisson arrival λT_f	1 packet/slot
Queue size K	10 packets
Nakagami parameter m	1 (Rayleigh fading)
Average SNR $\bar{\gamma}$	22 dB
Doppler frequency $f_d T_f$	0.02
Packet timeout P	4 slots
Target packet error rate P_0	from 10^{-4} to 10^{-1}

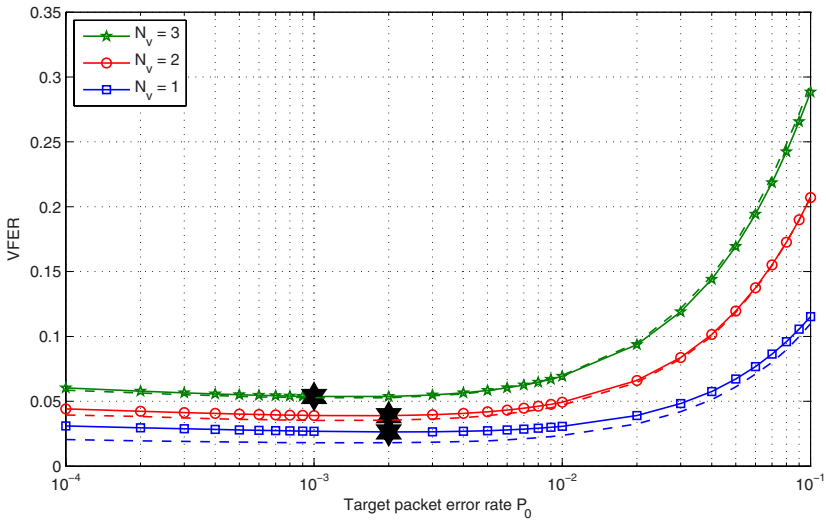


Fig. 7. VFER $\xi(N_v, P_0)$ versus P_0 with $P = 4$

different values of N_v . It is observed that the optimal P_0 is different for different N_v . To determine the optimal P_0 to use, the overall PSNR performance has to be considered.

From the analytical VFER curves of different N_v in Figure 7 the GOP distortion $D_{GOP}(P_0)$ and the resulting PSNR can be computed. The results are illustrate in Figure 8. The dashed lines and the solid lines nearly overlap which indicates our analysis is also accurate in capturing the PSNR performance. From the maximum point of the PSNR curves, we can find the optimal target PER P_0 for the video streaming. The SNR thresholds of the AMC can be determined accordingly. Note that the optimal P_0 of PSNR is not necessary the same with the optimal P_0 of VFER shown in the previous figure.

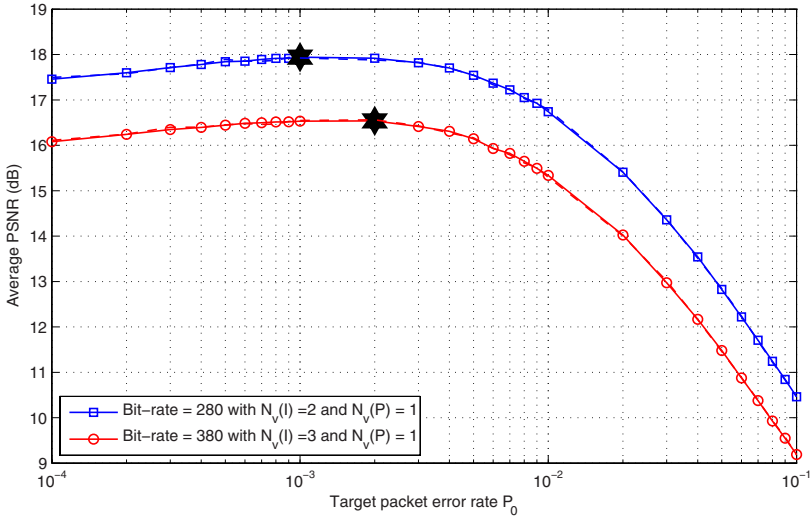


Fig. 8. Average PSNR in GOP versus P_0 with $P = 4$

6 Conclusions

In this paper, an analytical framework has been proposed to analyze the performance of streaming delay sensitive multimedia packets over the wireless fading channel employing the AMC transmission scheme at physical layer. Through the framework, the video frame error rate of video frames of different lengths has been derived. The GOP distortion and the associated PSNR have also been obtained from the video frame error rate. The PSNR can be expressed as a function of the target AMC PER. From the PSNR analysis, one can obtain the optimal target AMC PER to determine the optimal SNR thresholds for the AMC to support multimedia streaming. Simulations show that the proposed analysis is accurate in capturing the performance of the video streaming. It is useful for optimizing the performance of wireless multimedia streaming.

Acknowledgment

This work was supported in part by the Excellent Research Projects of National Taiwan University under Contract 97R0062-06, by the National Science Council under Contract NSC97-2220-E-002-017, by the Ministraton of Education.

References

1. Liu, Q., Zhou, S., Giannakis, G.B.: Queueing with Adaptive Modulation and Coding over Wireless Links: Cross-layer Analysis and Design. *IEEE Trans. Wireless Commun.* 4(3), 1142–1153 (2005)

2. Liu, Q., Zhou, S., Giannakis, G.B.: Analyzing and Optimizing Adaptive Modulation Coding Jointly with ARQ for QoS-guaranteed Traffic. *IEEE Trans. Veh. Technol.* 56(2), 710–720 (2007)
3. Zhou, S., Zhang, K., Niu, Z., Yang, Y.: Queueing Analysis on MIMO Systems with Adaptive Modulation and Coding. In: *IEEE Intl. Conf. Commun.* (2008)
4. Xu, J., Shen, X., Mark, J.W., Cai, J.: Adaptive Transmission of Multi-layered Video over Wireless Fading Channels. *IEEE Trans. Wireless Commun.* 6(6), 2305–2314 (2007)
5. Zhang, X., Tang, J., Chen, H., Ci, S., Guizani, M.: Cross-Layer-Based Modeling for Quality of Service Guarantees in Mobile Wireless Networks. *IEEE Commun. Mag.* (January 2006)
6. Harsini, J., Lahouti, F.: Adaptive Transmission Policy Design for Delay-Sensitive and Bursty Packet Traffic over Wireless Fading Channels. *IEEE Trans. Wireless Commun.* (2008)
7. Doufexi, A., Armour, S., Butler, M., Nix, A., Bull, D., McGeehan, J.: A Comparison of the HIPERLAN/2 and IEEE 802.11a Wireless LAN Standards. *IEEE Commun. Mag.* (May 2002)
8. Liu, Q., Zhou, S., Giannakis, G.B.: Cross-layer Combining of Adaptive Modulation and Coding with Truncated ARQ over Wireless Links. *IEEE Trans. Wireless Commun.* 3(5), 1746–1755 (2004)
9. Molisch, A.: *Wireless Communications*. John Wiley and Sons, Chichester (2005)
10. Alouini, M., Goldsmith, A.: Adaptive Modulation over Nakagami Fading Channels. *Wireless Personal Communications* 13, 119–143 (2000)
11. Razavilar, J., Liu, K.J., Marcus, S.: Jointly Optimized Bit-Rate/Delay Control Policy for Wireless Packet Networks with Fading Channels. *IEEE Trans. Commun.* 50(3), 484–494 (2002)
12. Yacoub, M., Bautista, J., de Rezende Guedes, L.: On Higher Order Statistics of the Nakagami-m Distribution. *IEEE Trans. Veh. Technol.* 48(3), 790–794 (1999)
13. Chou, P.A., Miao, Z.: Rate-Distortion Optimized Streaming of Packetized Media. *IEEE Trans. Multimedia* 8(2), 390–404 (2006)

Credit-Token Based Inter-cell Radio Resource Management: A Game Theoretic Approach

Chun-Han Ko and Hung-Yu Wei

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan
b91901127@ntu.edu.tw,
hywei@cc.ee.ntu.edu.tw

Abstract. In this paper, a radio resource sharing scheme for wireless cellular network is investigated to achieve efficiency and fairness among base stations. We propose a credit-token based spectrum sharing algorithm. Game theory is utilized to formulate and analyze the proposed spectrum sharing algorithm. We first discuss the simplest two-base-station game through a graphical method to gain insights for the solution. Afterwards, the Nash Equilibrium of the n -base-station game is derived and the spectrum allocation at the Nash equilibrium is shown to be unique. Several desirable properties, including allocative efficiency, Pareto optimality, weighted max-min fairness, and weighted proportional fairness, are proved to be attained at the Nash equilibrium. Furthermore, we design a strategy-proof spectrum allocation mechanism based on the proposed spectrum sharing algorithm so that truthful declarations of spectrum demands maximize the performance in each cell.

Keywords: wireless cellular network, spectrum sharing, credit token, game theory.

1 Introduction

Dynamic spectrum sharing has been a promising approach to increase the efficiency of spectrum usage [1]. In the realm of dynamic spectrum sharing, many researchers are interested in introducing pricing mechanisms to further achieve efficient and fair spectrum utilization [2,3,4]. Credit token is one of such possible pricing solutions. The concept of credit token and its utilization in dynamic spectrum sharing are first introduced in [4]. Credit token is similar to money except that credit token can be frozen but cannot be exchanged. In IEEE 802.22 standard, credit token is also used in the self-coexistence mechanism in the MAC protocol [5].

Recently, game theory has been applied to model dynamic spectrum sharing among BSs. S. Sengupta *et al.* applied minority game theory to investigate the problem that whether a BS should stay at the present channel or switch to another channel [6]. They showed a mixed strategy Nash equilibrium existed and the mixed strategy space performed better than the pure strategy space

in achieving optimal solution. D. Gao *et al.* modeled the dynamic renting and offering mechanism as a progressive second price auction [7]. The utilization of this auction mechanism had a major benefit that BSs would make their requests truthfully. D. Niyato *et al.* formulated the transaction of spectrum bands between licensed users and BSs by a sealed-bid double auction [8]. They also introduced a pricing mechanism to model the service between BSs and users. Nash equilibrium was found through a numerical method.

In this paper, we aim to find a game theoretic solution for inter-cell radio resource management. We propose a credit-token based spectrum sharing algorithm which comprises mechanisms of spectrum renting, offering, and contention. By applying game theory to formulate the spectrum sharing problem, we confirm that a Nash equilibrium always exists and the spectrum allocation at the Nash equilibrium is always unique. Several desirable properties, including allocative efficiency, Pareto optimality, weighted max-min fairness, and weighted proportional fairness, are attained at the Nash equilibrium. Finally, extended from the spectrum sharing algorithm, we devise a strategy-proof, efficient and fair spectrum allocation mechanism to adopt in the general case that BSs' max spectrum demands are private information.

2 Spectrum Sharing Scheme

2.1 System Model

The system we consider consists of an agent, A , and n BSs, BS_i for $i = 1, 2, \dots, n$. Agent A , serving as a marketplace, manages resource transactions among all BSs. Agent A also offers free spectrum using time O . (If O is less than zero, "offering O " means "retrieving $-O$.") Each BS_i has a single orthogonal spectrum band, spectrum using time T , a credit token budget B_i , and a max traffic demand x_i (in time) additional to T . All of these are assumed to be public information. Figure 1(a) is an illustration of a system of Agent A and three BSs. Figure 1(b) is the corresponding max additional traffic demands. The notations are summarized in Table 1. In the rest parts of the paper, we will use "spectrum" to denote spectrum using time for short.

2.2 Credit-Token Based Spectrum Sharing Algorithm

We propose a credit-token based spectrum sharing algorithm. The algorithm has two phases: a spectrum renting-and-offering phase and a spectrum contention phase. We assume each BS_i will use credit tokens for spectrum acquisition and spectrum protection.

Initially, Agent A broadcasts that the renting-and-offering phase starts with spectrum O provided. After hearing the broadcasting, each BS_i will make an acquisition/offering request, y_i , which is the spectrum it claims to acquire if $y_i > 0$ or to offer if $y_i < 0$. Each BS_i is accordingly referred to as an acquirer or an offeror. As each BS_i makes its spectrum request, an assumption is adopted that

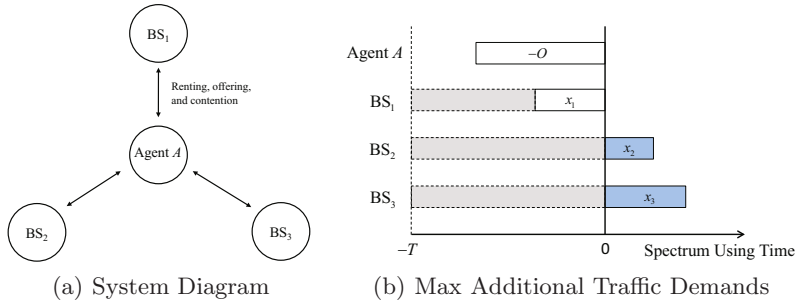


Fig. 1. System of Agent A and Three BSs

every unit of the spectrum BS_i wants to acquire, $[y_i]^+$, and of the spectrum BS_i wants to protect, $T - [-y_i]^+$, is equally important. Therefore the credit tokens should be fairly allocated. The unit spectrum acquisition price and the unit spectrum protection price are then both equal to $p_i(y_i) = \frac{B_i}{T+y_i}$, as depicted in Figure 2 (The function $[\cdot]^+$ gives a non-negative value.) Alternatively, as Agent A receives the acquisition/offering requests from all BSs, it collects the offered spectrum from the offerors and then assigns the collected spectrum and O to the acquirers, in decreasing order of the unit acquisition price, for their requested amount until exhaustion. When multiple acquirers have the same unit acquisition price and there is not enough spectrum for them, the spectrum assigned to them is assumed proportional to their requested amounts.

If the acquirers cannot get enough spectrum in the renting-and-offering phase, the contention phase starts. In the contention phase, Agent A first collects each BS_i 's spectrum to protect, $T - [-y_i]^+$. The collected $(T - [-y_i]^+)$ s are then sorted in increasing order of the unit protection price. Afterwards Agent A assigns the sorted $(T - [-y_i]^+)$ s to the acquirers for their inadequate amounts in decreasing order of the unit acquisition price. The assignment ends if the unit protection price is greater than or equal to the unit acquisition price. Finally, Agent A returns the unassigned spectrum back to original BSs. When multiple acquirers have the same acquisition price and there is not enough spectrum for them, we assume the spectrum assigned to them is proportional to their inadequate amounts. When multiple BSs have the same protection price and their spectrum is assigned to others, we assume the assigned spectrum is fairly afforded by these BSs.

After both renting-and-offering and contention phases finish, the credit tokens the acquirers spend for spectrum acquisition are frozen and data transmission begins. We show, in Table 1, the mathematical expressions of the spectrum BS_i acquires or offers in the renting-and-offering phase and the spectrum BS_i acquires or loses in the contention phase. The former is $\min(y_i, r_i)$ and the latter is $\min([y_i - r_i]^+, c_i)$. The total spectrum BS_i acquires or loses in both phases is therefore $\min(y_i, r_i) + \min([y_i - r_i]^+, c_i)$. However, we will use $\min(y_i, t_i)$ to

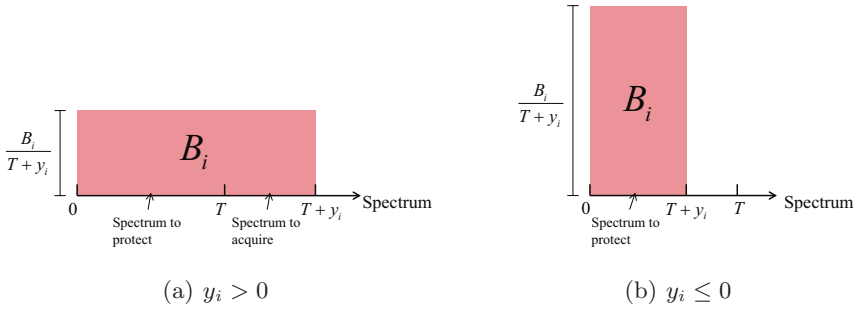


Fig. 2. Unit Spectrum Acquisition and Protection Price of BS_{*i*}

Table 1. Notations

T	Spectrum owned by each BS.
O	Spectrum offered by Agent A .
B_i	Credit token budget of BS _{<i>i</i>} .
x_i	Max traffic demand additional to T of BS _{<i>i</i>} .
y_i	Spectrum acquisition/offering request of BS _{<i>i</i>} .
$p_i(y_i)$	Unit spectrum acquisition and protection price of BS _{<i>i</i>} . $p_i(y_i) = \frac{B_i}{T+y_i}$
$\min(y_i, r_i)$	Spectrum BS _{<i>i</i>} acquires or offers in the renting-and-offering phase. $r_i(\mathbf{y}) = \frac{[y_i]^+}{\sum_{j:p_j=p_i} [y_j]^+} \left[O + \sum_{j=1}^n [-y_j]^+ - \sum_{j:p_j>p_i} [y_j]^+ \right]^+$
$\min([y_i - r_i]^+, c_i)$	Spectrum BS _{<i>i</i>} acquires or loses in the contention phase. $c_i(\mathbf{y}) = \frac{[y_i - r_i]^+}{\sum_{j:p_j=p_i} [y_j - r_j]^+} \left[\sum_{j:p_j<p_i} (T - [-y_j]^+) - \sum_{j:p_j>p_i} [y_j - r_j]^+ \right]^+ - (T - [-y_i]^+)$ $+ \left[(T - [-y_i]^+) - \frac{T - [-y_i]^+}{\sum_{j:p_j=p_i} (T - [-y_j]^+)} \left[- \sum_{j:p_j<p_i} (T - [-y_j]^+) + \sum_{j:p_j>p_i} [y_j - r_j]^+ \right]^+ \right]^+$
$\min(y_i, t_i)$	Spectrum BS _{<i>i</i>} acquires or loses in both phases. $\min(y_i, t_i) = \min(y_i, r_i) + \min([y_i - r_i]^+, c_i)$ $t_i(\mathbf{y}) = \frac{[y_i]^+}{\sum_{j:p_j=p_i} [y_j]^+} \left[O + \sum_{j:p_j=p_i} [y_j]^+ - \sum_{j:p_j \geq p_i} y_j + \sum_{j:p_j < p_i} T \right]^+ - T$ $+ \left[(T - [-y_i]^+) - \frac{T - [-y_i]^+}{\sum_{j:p_j=p_i} (T - [-y_j]^+)} \left[-O - \sum_{j:p_j=p_i} [y_j]^+ + \sum_{j:p_j \geq p_i} y_j - \sum_{j:p_j < p_i} T \right]^+ \right]^+$
$P_i(\mathbf{y})$	Frozen credit tokens of BS _{<i>i</i>} . $P_i(\mathbf{y}) = p_i(y_i) [\min(y_i, t_i)]^+$

represent the total spectrum BS_i acquires or loses in both phases for simplicity. (Due to lack of space, we skip the proof here.)

3 Game Formulation

The problem we want to study is as follows.

Problem. *Given that the original spectrum T , the credit token budget B_i , and the max traffic demand x_i of each BS_i are public information, if the acquisition/offering request y_i is such that $-T \leq y_i \leq x_i$, how does each BS_i make the acquisition/offering request to increase the spectrum?*

From each BS’s perspective, spectrum sharing is intrinsically a game that each BS unitarily optimizes its performance by acquiring or offering spectrum. We utilize game theory to find if there is any steady state, Nash equilibrium, for this the spectrum sharing problem. Game theory is a set of mathematical tools for analyzing interactive decision processes [9]. Three primary components comprise a game: a player set N ; a strategy space $S = \prod_{i \in N} S_i$ where $S_i, i \in N$ is player i strategy set; a utility-function set $U = \{u_i(\mathbf{s})\}$, where $u_i(\mathbf{s}), i \in N$ is player i ’s utility under a strategy profile $\mathbf{s} \in S$. In a game, a steady state where no player will unitarily deviate is called a Nash equilibrium [10].

Table 2. Spectrum Sharing Game Model

$G = (N, Y, U, B, X)$	
Player Set N	$N = \{1, 2, \dots, n\}$. BSs are the players of the game.
Strategy Space Y	$Y = \prod_{i \in N} Y_i$ and $Y_i = \{y_i : -T \leq y_i \leq x_i\} \forall i \in N$. We treat BS_i ’s acquisition/offering request y_i as the strategy.
Utility-function Set U	$U = \{u_i(\mathbf{y})\}$ and $u_i(\mathbf{y}) = \min(y_i, t_i) \forall i \in N$. Since each BS aims to increase its spectrum, it is reasonable to set the spectrum as the utility. We do not include any pricing term because each BS never receives credit tokens. (Recall that credit tokens can only be frozen.) We also ignore the constant term T for convenience. Each BS’s utility is therefore the spectrum it acquires or loses from renting, offering, and contention.
Credit-token-budget Set B	$B = \{B_i\}$
Max Traffic Set X	$X = \{x_i\}$ with $\{p_i(x_i)\}$ in decreasing order. Without losing generality, we assume $\{p_i(x_i)\}$ is sorted in decreasing order.

Definition 1. A strategy profile $\mathbf{s}^* = (s_i^*, s_{-i}^*)$ is a Nash equilibrium if

$$u_i(\mathbf{s}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \neq s_i^* \text{ and } \forall i \in N$$

The best response function of any player depicts his best (in term of highest utility) strategy given all possible s_{-i} from other players. A Nash equilibrium can also be defined by best response functions.

Definition 2. $BR_i(s_{-i})$ is the best response function of player i if

$$BR_i(s_{-i}) = \{s_i : u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}), \forall s'_i \neq s_i\}$$

Definition 3. A strategy profile $\mathbf{s}^* = (s_i^*, s_{-i}^*)$ is a Nash equilibrium if

$$s_i^* = BR_i(s_{-i}^*) \quad \forall i \in N$$

By applying game theory, we construct a game model, denoted as G , for the spectrum sharing problem. The game model is shown in Table 2 with each BS’s credit token budget and max traffic demand taken into account.

4 Graphical Analysis – Two Players with Same Budget

To gain insights for the solution of the general n -player game, we derive the Nash equilibrium in the simplest 2-same-budget-player game through a graphical method. We draw both players’ best response functions together. The resulting intersection is the Nash equilibrium. Recall we assume $p_1(x_1) \geq p_2(x_2)$. When both players have the same credit token budget, this assumption reduces to $x_1 \leq x_2$. Accordingly, the system traffic demands can be classified into three cases: $x_1 \leq \frac{O}{2}$ and $x_2 \leq O - x_1$; $x_1 \leq \frac{O}{2}$ and $x_2 > O - x_1$; $x_1 > \frac{O}{2}$ and $x_2 > \frac{O}{2}$.

4.1 Traffic Case 1 - $x_1 \leq \frac{O}{2}$ and $x_2 \leq O - x_1$

As illustrated in Figure 3(a), the best response function of player 1 is uniquely x_1 . It means player 1 will always play the unique dominant strategy, $y_1 = x_1$. We call this strategy a dominant one since it always results in higher utility than all other strategies. Also, player 2’s best response function is x_2 . Player 2 plays the unique dominant strategy, $y_2 = x_2$. The intersection of two best response functions is (x_1, x_2) , a unique Nash equilibrium. The corresponding utility profile is (x_1, x_2) as well.

4.2 Traffic Case 2 - $x_1 \leq \frac{O}{2}$ and $x_2 > O - x_1$

We already know player 1 plays the unique dominant strategy, $y_1 = x_1$, when $x_1 \leq \frac{O}{2}$. In Figure 3(b), player 2’s best response function is $BR_2(y_1) = O - y_1 \sim x_2$ which implies that the strategy, $y_2 = x_2$, is player 2’s unique dominant strategy. However, it is not meaningful to discuss the concept of dominant strategy

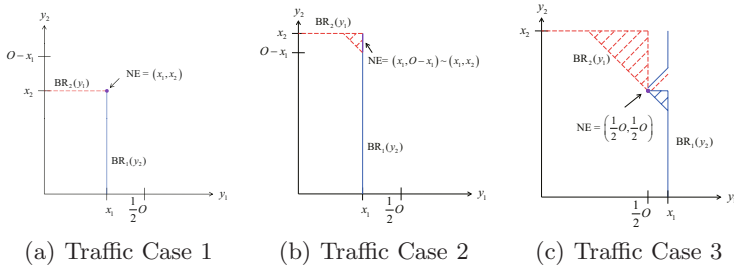


Fig. 3. Best Response Functions and Nash Equilibrium

for player 2 while it is like to play a single-player game. We explain why player 2 is like to play a single-player game. When $x_1 \leq \frac{O}{2}$, player 1 plays the unique dominant strategy, $y_1 = x_1$, and acquires x_1 from O . (When x_1 is less than zero, “acquiring x_1 ” means “offering $-x_1$.”) For player 2, it has $(O - x_1)$ remained to acquire without any other player. Therefore player 2 is like to play a single-player game and it can always acquire $(O - x_1)$ by playing y_2 such that $O - x_1 \leq y_2 \leq x_2$. This fact obviously results in multiple Nash equilibria. This can also be shown from the intersection of two best response functions, a line segment between $(x_1, O - x_1)$ and (x_1, x_2) . It means multiple Nash equilibria, $(x_1, O - x_1 \sim x_2)$, exist. Though multiple Nash equilibria exist, the corresponding utility profile is uniquely $(x_1, O - x_1)$.

4.3 Traffic Case 3 - $x_1 > \frac{O}{2}$ and $x_2 > \frac{O}{2}$

In Figure 3(c), the best response function of player 1 is

$$BR_1(y_2) = \begin{cases} \min(O - y_2, x_1) \sim x_1 & \text{if } y_2 \leq \frac{O}{2} \\ y_2^- & \text{if } \frac{O}{2} < y_2 \leq x_1 \\ x_1 & \text{if } x_1 < y_2 \end{cases}$$

and the best response function of player 2 is

$$BR_2(y_1) = \begin{cases} \min(O - y_1, x_2) \sim x_2 & \text{if } y_1 \leq \frac{O}{2} \\ y_1^- & \text{if } \frac{O}{2} < y_1 \end{cases}$$

We see neither player 1 nor player 2 has dominant strategy. The intersection of two best response functions is $(\frac{O}{2}, \frac{O}{2})$, a unique equal-strategy Nash equilibrium. The corresponding utility profile is $(\frac{O}{2}, \frac{O}{2})$.

We summarize the observations as follows. These observations, playing the essential roles in the two-same-budget-player game, can be extended in the general n -different-budget-player game.

1. Condition for unique dominant strategies: When $x_1 \leq \frac{O}{2}$, player 1 plays the unique dominant strategy, $y_1 = x_1$. When $x_2 \leq O - x_1$, player 2 plays the unique dominant strategy, $y_2 = x_2$.

2. Existence of a Nash equilibrium: A Nash equilibrium always exists in all cases.
3. Condition for multiple Nash equilibria: The only case where multiple Nash equilibria exist is $x_1 \leq \frac{O}{2}$ and $x_2 > O - x_1$. We have explained that because player 2 is like to play a single-player game with $(O - x_1)$ offered, it can always acquire $(O - x_1)$ by playing $O - x_1 \leq y_2 \leq x_2$. Multiple Nash equilibria, $(x_1, O - x_1 \sim x_2)$, hence exist.
4. Unique utility profile at the Nash equilibrium: Even in the multi-Nash-equilibrium case, the corresponding utility profile is unique.

5 Mathematical Analysis – n Players

In this section, we first extend the two-same-budget-player game to the general n -different-budget-player game, i.e. Game G . The extension is summarized in Table 3. Afterwards, we do formal derivations for the Nash equilibrium of Game G .

Table 3. Summary of Extension

	2-Same-Budget-Player Game	n -Different-Budget-Player Game
Traffic Threshold	$\{\frac{O}{2}, O - x_1\}$	$\{e_{j, -(j-1)}\}$
Traffic Case	$x_1 > \frac{O}{2}$ and $x_2 > \frac{O}{2}$; $x_1 \leq \frac{O}{2}$ and $x_2 > O - x_1$; $x_1 \leq \frac{O}{2}$ and $x_2 \leq O - x_1$	$x_j \leq e_{j, -(j-1)} \forall j \in \{1, \dots, k\}$, $x_j > e_{j, -k} \forall j \in \{k + 1, \dots, n\}$ where $k \in \{0, N\}$
Nash Equilibrium	$(\frac{O}{2}, \frac{O}{2})$; $(x_1, O - x_1 \sim x_2)$; (x_1, x_2)	$(x_1, \dots, x_k, e_{k+1}, \dots, e_{n, -k})$ if $k \neq n - 1$; $(x_1, \dots, x_{n-1}, e_{n, -(n-1)} \sim x_n)$ if $k = n - 1$

5.1 Extension from Two-Player Game to n -Player Game

Recall that we have assumed the max traffic demands are such that $\{p_i(x_i)\}$ is ranged in decreasing order. In the two-same-budget-player game, we see there are two traffic thresholds, $\frac{O}{2}$ and $O = x_1$. Accordingly, the traffic can be categorized into three cases: $x_1 > \frac{O}{2}$ and $x_2 > \frac{O}{2}$; $x_1 \leq \frac{O}{2}$ and $x_2 > O - x_1$; $x_1 \leq \frac{O}{2}$ and $x_2 \leq O - x_1$. The corresponding Nash equilibrium is $(\frac{O}{2}, \frac{O}{2})$, $(x_1, O - x_1 \sim x_2)$, and (x_1, x_2) .

Extended from the two-same-budget-player game, it is reasonably to guess the n - same-budget-player game has the set of n traffic thresholds, $\left\{ \frac{-\sum_{l=0}^{j-1} x_l}{n-j+1} \right\}$, where $x_0 = -O$. To further extend to the n -different-budget-player game, we must know what plays the same role as $\frac{-\sum_{l=0}^k x_l}{n-k}$ in the n -same-budget-player game.

Definition 4. For Game G , we define

$$e_{j,-k} \equiv \frac{B_j}{\frac{1}{n-k} \sum_{l=k+1}^n B_l} \left(\frac{-\sum_{l=0}^k x_l}{n-k} \right) + \left(\frac{B_j}{\frac{1}{n-k} \sum_{l=k+1}^n B_l} - 1 \right) T$$

$\forall j \in \{k+1, \dots, n\}$ and $\forall k \in \{0, N\}$, where $x_0 = -O$

$e_{j,-k}$ can be interpreted as weighted and translated $\frac{-\sum_{l=0}^k x_l}{n-k}$ with the weight $\frac{B_j}{\frac{1}{n-k} \sum_{l=k+1}^n B_l}$. The term $-k$ in the subscript indicates that player i , $i \in \{1, \dots, k\}$, which has already acquired x_i from O , is excluded. When $k = 0$, $e_{j,-0}$ is denoted as e_j for short. Following the definition, there is a corollary stating some properties of $e_{j,-k}$.

Corollary 1. For Game G , the following statements about $e_{j,-k}$ are always true:

1. $p_j(e_{j,-k}) = \frac{\frac{1}{n-k} \sum_{l=k+1}^n B_l}{T + \frac{-\sum_{l=0}^k x_l}{n-k}} \forall j \in \{k+1, \dots, n\}$.
2. $\sum_{j=1}^k x_j + \sum_{j=k+1}^n e_{j,-k} = \min \left(O, \sum_{j=1}^n x_j \right) \forall k \in \{0, N\}$.
3. $x_k \leq e_{k,-(k-1)} \Leftrightarrow x_j \leq e_{j,-(j-1)} \forall j \in \{1, \dots, k\}$.
4. $x_{k+1} > e_{k+1,-k} \Leftrightarrow x_j > e_{j,-k} \forall j \in \{k+1, \dots, n\}$.
5. $x_k \leq e_{k,-(k-1)} \Rightarrow p_k(e_{k,-(k-1)}) \geq p_j(e_{j,-k}) \forall j \in \{k+1, \dots, n\}$.

When $B_i = B_j \forall i, j \in N$, the weights for all $e_{j,-k}$ become 1 and $e_{j,-k}$ reduces to $\frac{-\sum_{l=0}^k x_l}{n-k}$. It is intuitively to believe that $e_{j,-k}$ play the same roles as $\frac{-\sum_{l=0}^k x_l}{n-k}$ in the same-budget case. Hence Game G should have the set of n traffic thresholds, $\{e_{j,-(j-1)}\}$. Besides, we can classify the traffic into $(n+1)$ cases where the $(k+1)$ -th case, $k \in \{0, N\}$, is $x_k \leq e_{k,-(k-1)}$ and $x_{k+1} > e_{k+1,-k}$. From Corollary 13 and 14, the $(k+1)$ -th case can equivalently represented as $x_j \leq e_{j,-(j-1)} \forall j \in \{1, \dots, k\}$ and $x_j > e_{j,-k} \forall j \in \{k+1, \dots, n\}$.

Definition 5. For Game G and $\forall k \in \{0, N\}$, we define

$$Traffic_k \equiv x_j \leq e_{j,-(j-1)} \quad \forall j \in \{1, \dots, k\} \text{ and } x_j > e_{j,-k} \quad \forall j \in \{k+1, \dots, n\}$$

The Nash equilibrium under $Traffic_k$ should be $(x_1, \dots, x_k, e_{k+1}, \dots, e_{n,-k})$ if $k \neq n-1$, and $(x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$ if $k = n-1$.

5.2 *n*-Player Game

Before starting, we should mention that we will use u_i and t_i to express $u_i(\mathbf{y})$ and $t_i(\mathbf{y})$ at any given strategy profile \mathbf{y} for short. If we need to compare the results between two different strategy profiles, say (y_i, y_{-i}) and (y'_i, y_{-i}) , we will distinguish by using u'_i and t'_i to express $u_i(y'_i, y_{-i})$ and $t_i(y'_i, y_{-i})$. Also, due to lack of space, we will only give proofs of important theorems.

First, Lemma 1 reveals the increasing property of utility functions with respect to strategies.

Lemma 1. *Given that Game G is under $Traffic_k$, $k \in \{0, N\}$, the following statements are always true:*

1. $u_i = y_i \ \forall i \in \{1, \dots, k\}$.
2. if $y_i \leq e_{i,-k}$ for some $i \in \{k + 1, \dots, n\}$, $u_i = y_i$.

Lemma 1 shows that $u_i, i \in \{1, \dots, k\}$, is an increasing function of y_i under $Traffic_k$. Therefore player i can always play $y_i = x_i$ to get the highest utility. In words, player $i, i \in \{1, \dots, k\}$, plays the unique dominant strategy, $y_i = x_i$.

Theorem 1. *Given that Game G is under $Traffic_k$, $k \in \{0, N\}$, player $i, i \in \{1, \dots, k\}$, plays the unique dominant strategy, $y_i = x_i$.*

Recall we have guessed the Nash equilibrium under $Traffic_k$ is $(x_1, \dots, x_k, e_{k+1}, \dots, e_{n,-k})$ if $k \neq n - 1$ and $(x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$ if $k = n - 1$. To verify our guess is correct, we prove that all other strategy profiles cannot be a Nash equilibrium. The proof is taken into two parts. The first part is to show that $y_i < x_i$ for any $i \in \{1, \dots, k\}$ or $y_i < e_{i,-k}$ for any $i \in \{k + 1, \dots, n\}$ is not in any Nash equilibrium. The other part is to show that $y_i > e_{i,-k}$ for any $i \in \{k + 1, \dots, n\}$ is not in any Nash equilibrium.

Lemma 2. *For Game G under $Traffic_k$, $k \in \{0, N\}$, $y_i < x_i$ for any $i \in \{1, \dots, k\}$ or $y_i < e_{i,-k}$ for any $i \in \{k + 1, \dots, n\}$ is not in any Nash equilibrium.*

Lemma 3. *For Game G under $Traffic_k$, $k \in \{0, N\}$ and $k \neq n - 1$, $y_i > e_{i,-k}$ for any $i \in \{k + 1, \dots, n\}$ is not in any Nash equilibrium.*

Combining Lemma 2 and Lemma 3, we have verified that under $Traffic_k$, any strategy profile other than $(x_1, \dots, x_k, e_{k+1}, \dots, e_{n,-k})$ if $k \neq n - 1$ and $(x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$ if $k = n - 1$ cannot be a Nash equilibrium. In words, only $(x_1, \dots, x_k, e_{k+1}, \dots, e_{n,-k})$ if $k \neq n - 1$ and $(x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$ if $k = n - 1$ can be a Nash equilibrium. We therefore check its property and find it a Nash equilibrium.

Theorem 2. *Given $Traffic_k$, $k \in \{0, N\}$ and $k \neq n - 1$, Game G has the unique Nash equilibrium, $NE_k = (x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$.*

Proof. Given $Traffic_k$, if Game G is at $(x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$, the corresponding utility profile is also $(x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$. For player i , $i \in \{1, \dots, k\}$, if it plays $y'_i < x_i$, then $u'_i = y'_i < u_i$. For player i , $i \in \{k+1, \dots, n\}$, if it plays $y'_i < e_{i,-k}$, $u'_i = y'_i < u_i$; if it plays, $y'_i > e_{i,-k}$, $u'_i = e_{i,-k}$. Consequently, $(x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$ meets the definition of Nash equilibrium. Since there is no other possible Nash equilibrium, Game G under $Traffic_k$, $k \in \{0, N\}$ and $k \neq n-1$, has the unique Nash equilibrium $(x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$. \square

Theorem 3. *Given $Traffic_{n-1}$, Game G has multiple Nash equilibria, $NE_{n-1} = (x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$.*

Proof. When $k = n-1$, it is proved in Theorem \square that player i , $i \in \{1, \dots, n-1\}$, plays the unique dominant strategy $y_i = x_i$. For player n , it is like to play a single-player game with $-\sum_{j=1}^{n-1} x_j$, equivalently $e_{n,-(n-1)}$, offered. Player n can play $e_{n,-(n-1)} \leq y_n \leq x_n$ such that $u_n = e_{n,-(n-1)}$. Hence G has multiple Nash equilibria, $NE_{n-1} = (x_1, \dots, x_{n-1}, e_{n,-(n-1)} \sim x_n)$. \square

After deriving the Nash equilibrium, we can easily verify that the utility profile at the Nash equilibrium is always unique. This is drawn by substituting all NE_k s into the utility functions.

Theorem 4. *Given $Traffic_k$, $k \in \{0, N\}$, Game G has the unique utility profile, $U_k^* = (x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$, at the Nash equilibrium NE_k .*

Recall we have set spectrum as utilities for all BSs. In system meaning, the utility profile at the Nash equilibrium represents the spectrum allocation at the Nash equilibrium. Theorem \square in words, reveals that our spectrum sharing algorithm always results in the unique traffic-dependent spectrum allocation at the Nash equilibrium, $AR^* = U_k^*$ given $Traffic_k$, $k \in \{0, N\}$.

6 Properties at Nash Equilibrium

After deriving the Nash equilibrium, we can proof that the spectrum allocation at the Nash equilibrium meets the criteria of allocative efficiency, Pareto optimality, weighted max-min fairness, and weighted proportional fairness.

Allocative efficiency \square means that a resource allocation maximizes total utilities over all players. It is regarded as the most optimality since no other allocations can achieve greater social welfare. Pareto optimality \square is defined as an allocation upon which no player can be made happier (in utility) without making at least one other player less happy. It is true that allocative efficiency always implies Pareto optimality. The mathematical definitions of allocative efficiency and Pareto optimality are given as below. To conform with the expressions in our game, we use \mathbf{y} and Y instead of \mathbf{s} and S to represent the strategy profile and the strategy space respectively.

Definition 6. A resource allocation game is allocatively efficient if the Nash equilibrium is a solution to the optimization problem

$$\max \sum_{i=1}^n u_i(\mathbf{y}) \quad \text{s.t. } \mathbf{y} \in Y$$

Definition 7. A resource allocation game is Pareto optimal if the Nash equilibrium \mathbf{y}^* satisfies

$$\exists \mathbf{y}' \neq \mathbf{y}^*, u_i(\mathbf{y}') > u_i(\mathbf{y}^*) \Rightarrow \exists j \in N, u_j(\mathbf{y}') < u_j(\mathbf{y}^*)$$

An allocation satisfies weighted max-min fairness [12] if it is not possible to increase one player’s weighted utility without simultaneously decreasing another player’s weighted utility which is already smaller. An allocation exhibits weighted proportional fairness [12] if it maximizes the product of all players’ utilities with weights in exponents.

Definition 8. A resource allocation game is weighted max-min fair with the weights $\{w_i\}$, if the Nash equilibrium is a solution to the optimization problem

$$\max \min \left(\frac{u_1(\mathbf{y})}{w_1}, \dots, \frac{u_n(\mathbf{y})}{w_n} \right) \quad \text{s.t. } \mathbf{y} \in Y$$

Definition 9. A resource allocation game is weighted proportional fair with the weights $\{w_i\}$, if the Nash equilibrium is a solution to the optimization problem

$$\max \prod_{i=1}^n u_i(\mathbf{y})^{w_i} \quad \text{s.t. } \mathbf{y} \in Y$$

Recall we ignore the constant term T when setting spectrum as utilities. While discussing weighted max-min fairness and weighted proportional fairness, we should replace u_i with $(T + u_i) \forall i \in N$; otherwise, the objective functions will not be correctly characterized. We choose $\hat{B}_i = \frac{B_i}{\frac{1}{n} \sum_{j=1}^n B_j}$ as the weight for each

player i . This is because B_i , mainly influencing player i ’s priority to acquire and to protect spectrum in system meaning, is the power to increase player i ’s utility. Also, by showing the range of utility functions in Lemma 4, we can transform the constraints of the optimization problems above from strategy domain into utility domain. Consequently, we can prove the properties by verifying that the utility profile at the Nash equilibrium is a solution to the corresponding optimization problems. We prove the properties of allocative efficiency and weighted max-min fairness here.

Lemma 4. For game G and $\forall \mathbf{y} \in Y$, the following statements about utility functions are always true:

1. $-T \leq u_i(\mathbf{y}) \leq x_i \forall i \in N$.
2. $-nT \leq \sum_{i=1}^n u_i(\mathbf{y}) \leq \min\left(O, \sum_{i=1}^n x_i\right)$.

Theorem 5. *Game G is allocatively efficient. Equivalently, the utility profile at the Nash equilibrium is a solution to the optimization problem,*

$$\max \sum_{i=1}^n u_i \quad \text{s.t.} \quad -T \leq u_i \leq x_i \forall i \in N \text{ and } -nT \leq \sum_{i=1}^n u_i \leq \min\left(O, \sum_{i=1}^n x_i\right)$$

Proof. Recall in Theorem 4 that the utility profile under $Traffic_k$, $k \in \{0, N\}$, is $U_k^* = (x_1, \dots, x_k, e_{k+1, -k}, \dots, e_{n, -k})$. Corollary 112 shows $\sum_{i=1}^k x_i + \sum_{i=k+1}^n e_{i, -k} = \min\left(O, \sum_{i=1}^n x_i\right) \forall k \in \{0, N\}$. Therefore we know $\sum_{i=1}^n u_i$ is maximized by $U_k^* \forall k \in \{0, N\}$. Game G is allocatively efficient. \square

Theorem 6. *Game G is weighted max-min fair with the weights $\{\hat{B}_i\}$. Equivalently, the utility profile at the Nash equilibrium is a solution to the optimization problem,*

$$\begin{aligned} & \max \min\left(\frac{T + u_1}{\hat{B}_1}, \dots, \frac{T + u_n}{\hat{B}_n}\right) \\ & \text{s.t.} \quad -T \leq u_i \leq x_i \forall i \in N \text{ and } -nT \leq \sum_{i=1}^n u_i \leq \min\left(O, \sum_{i=1}^n x_i\right) \end{aligned}$$

Proof. When Game G is under $Traffic_0$, by substituting U_0^* into the objective function and using Corollary 111, we derive

$$\frac{T + u_i}{\hat{B}_i} = \frac{T + e_i}{\hat{B}_i} = \frac{T + \frac{O}{n}}{\frac{1}{n} \sum_{l=1}^n B_l} = T + \frac{O}{n} \quad \forall i \in N \tag{1}$$

$$\min\left(\frac{T + u_1}{\hat{B}_1}, \dots, \frac{T + u_n}{\hat{B}_n}\right) = \min\left(T + \frac{O}{n}, \dots, T + \frac{O}{n}\right) = T + \frac{O}{n} \tag{2}$$

Because $\sum_{i=1}^n e_i = O$, if $u_j > e_j$ for some player j , there must be some player m having $u_m < e_m$. Therefore we have

$$\min\left(\frac{T + u_1}{\hat{B}_1}, \dots, \frac{T + u_n}{\hat{B}_n}\right) < \min\left(\dots, \frac{T + e_m}{\hat{B}_m}, \dots\right) \leq T + \frac{O}{n} \tag{3}$$

Equation (3) tells that $\min\left(\frac{T+u_1}{\hat{B}_1}, \dots, \frac{T+u_n}{\hat{B}_n}\right)$ is maximized by U_0^* .

When Game G is under $Traffic_k$ where $k \neq 0$, we know, from Corollary [III.5](#), $p_k(e_{k,-(k-1)}) \geq p_j(e_{j,-k}) \forall j \in \{k+1, \dots, n\}$. Then $p_1(x_1) \geq \dots \geq p_k(x_k) \geq p_k(e_{k,-(k-1)}) \geq p_j(e_{j,-k}) \forall j \in \{k+1, \dots, n\}$. This is equivalent to

$$\frac{T + x_1}{\hat{B}_1} \leq \dots \leq \frac{T + x_k}{\hat{B}_k} \leq \frac{T + e_{k,-(k-1)}}{\hat{B}_k} \leq \frac{T + e_{j,-k}}{\hat{B}_j} \quad \forall j \in \{k+1, \dots, n\} \quad (4)$$

Equation [\(4\)](#) reveals that the max value of $\min(\frac{T+u_1}{\hat{B}_1}, \dots, \frac{T+u_n}{\hat{B}_n})$ is $\frac{T+x_1}{\hat{B}_1}$ and is reached at $u_1 = x_1$. Since $u_1 = x_1$ is implied by U_k^* , $k \neq 0$, $\min(\frac{T+u_1}{\hat{B}_1}, \dots, \frac{T+u_n}{\hat{B}_n})$ is maximized by U_k^* where $k \neq 0$.

In summary, $\min(\frac{T+u_1}{\hat{B}_1}, \dots, \frac{T+u_n}{\hat{B}_n})$ is maximized by the utility profile at the Nash equilibrium. Game G is weighted max-min fair. □

7 Strategy-Proof Mechanism – Max Traffic Declaration

In the previous content, we already show the Nash equilibrium, the spectrum allocation result, and the corresponding properties of our game. If we omit the process of players’ making acquisition/offering requests and directly adopt the final spectrum allocation result, the proposed spectrum sharing algorithm can be simplified as the following spectrum allocation rule.

Definition 10. *Given that the spectrum T , the credit token budget B_i , and the max traffic demand x_i of each player i are public information and assuming that $\{p_i(x_i)\}$ is ranged in decreasing order without losing generality, the spectrum allocation is $AR^* = (x_1, \dots, x_k, e_{k+1,-k}, \dots, e_{n,-k})$ under $Traffic_k$, $k \in \{0, N\}$.*

According to this spectrum allocation rule, we can design a mechanism M to adopt in a more general case that all players’ max traffic demands are private information. In Mechanism M , each player i declares its max traffic demand, x'_i , which may be different from the true max traffic demand x_i . Given all players’ declarations, Mechanism M applies the spectrum allocation rule to allocate spectrum. Since now each player possibly gains more spectrum than its true max traffic demand, it is reasonable to add the assumption that when a player has reached its true max traffic demand, its utility is the true max traffic. Mechanism M is strategy-proof [III.3](#), i.e. the truth-revelation of the max traffic is a dominant-strategy equilibrium.

Theorem 7. *Mechanism M is strategy-proof. Equivalently, the strategy profile, (x_1, \dots, x_n) , is a dominant-strategy equilibrium.*

Proof. Given any x'_{-i} , we want to prove $x'_i = x_i$ always results in the highest utility for every player i under all traffic cases.

Let $\underline{N} = \{\underline{i}\}$ be the sorted player set N such that $\{p_{\underline{i}}(x'_{\underline{i}})\}$ is in decreasing order. Let $e_{\underline{j},-\underline{k}}$, $\forall \underline{k} \in \{0, \underline{N}\}$ and $\forall \underline{j} \in \{\underline{k}+1, \dots, \underline{n}\}$, be the same as $e_{j,-k}$ in

Definition 4 with $\{x_i\}$ replaced by $\{x'_i\}$. Also, let $Traffic_{\underline{k}}, \underline{k} \in \{0, \underline{N}\}$, denote $x'_j \leq e_{j, -(j-1)} \forall j \in \{\underline{1}, \dots, \underline{k}\}$ and $x'_j > e_{j, -k} \forall j \in \{\underline{k} + 1, \dots, \underline{n}\}$.

Assume that player i now plays $x'_i = x_i$ and has the m -th priority, i.e. $i = \underline{m}$ and $x_i = x'_m$. When $\underline{m} \leq \underline{k}$, we have $x_i = x'_m \leq e_{m, -(m-1)}$. Correspondingly, $u_i = x_i$ which is the highest utility player i can obtain. When $\underline{m} > \underline{k}$, we have $x_i = x'_m > e_{m, -k}$ and $u_i = e_{m, -k}$. If player i plays $x'_i < e_{m, -k}$, then $u'_i = x'_i < e_{m, -k} = u_i$; if player i plays $x'_i \geq e_{m, -k}$, then $u'_i = e_{m, -k}$. In words, no other strategy results in higher utility. From the above, $x'_i = x_i$ results in the highest utility under all traffic cases and therefore is a dominant strategy of player i .

Because the derivation above is applicable $\forall i \in N$, $x'_i = x_i$ is a dominant strategy of every player i and the strategy profile, (x_1, \dots, x_n) , is a dominant-strategy equilibrium. □

Given that Mechanism M is at (x_1, \dots, x_n) , the spectrum allocation result is the same as that in Theorem 4. Efficiency and fairness thus hold.

8 Conclusions

In this paper, we propose an efficient and fair spectrum sharing scheme. We show all BSs always reach a Nash equilibrium where the spectrum allocation is unique. The proposed spectrum sharing algorithm is desirable because it achieves efficiency and fairness among all BSs. The spectrum allocation is efficient as allocative efficiency and Pareto optimality are achieved. It also meets both weighted max-min fair and weighted proportional fair criteria. By adopting this spectrum allocation result, a strategy-proof mechanism, ensuring efficiency and fairness at the truth-revealing dominant-strategy equilibrium, is designed to apply in the more general case that max traffic demands are private information.

References

1. Akyildiz, I., Lee, W., Vuran, M., Mohanty, S.: Next Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: a Survey. Elsevier Computer Networks 50, 2127–2159 (2006)
2. Ji, Z., Liu, K.J.R.: Dynamic Spectrum Sharing: a Game Theoretical Overview. IEEE Communications Magazine, 88–94 (May 2007)
3. Ileri, O., Samardzija, D., Mandayam, N.B.: Demand Responsive Pricing and Competitive Spectrum Allocation via Spectrum Server. In: IEEE DySPAN 2005, November 2005, pp. 194–202 (2005)
4. Grandblaise, D., Moessner, K., Vivier, G., Tafazolli, R.: Credit Token based Rental Protocol for Dynamic Channel Allocation. In: 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications (2006)
5. IEEE P802.22/D1.0 Draft Standard for Wireless Regional Area Networks Part22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands (April 2008)

6. Sengupta, S., Chandramouli, R., Brahma, S., Chatterjee, M.: A Game Theoretic Framework for Distributed Self-Coexistence Among IEEE 802.22 Networks. In: IEEE Global Telecommunications Conference, IEEE GLOBECOM 2008, November 2008, pp. 1–6 (2008)
7. Gao, D., Cai, J., Li, Z., Wei, X., Chen, R.: Credit Token Based Dynamic Resource Renting and Offering Mechanism for Cognitive Radio WRAN BS Spectrum Sharing. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops, AINAW 2008, March 2008, pp. 296–300 (2008)
8. Niyato, D., Hossain, E., Han, Z.: Dynamic Spectrum Access in IEEE 802.22-Based Cognitive Wireless Networks: a Game Theoretic Model for Competitive Spectrum Bidding and Pricing. *IEEE Wireless Communications* 16(2), 16–23 (2009)
9. Osborne, M.J.: *An Introduction to Game Theory*. Oxford University Press, Oxford (2003)
10. Nash, J.F.: Equilibrium Points in n -Person Games. *Proceedings of the National Academy of Sciences* 36(1), 48–49 (1950)
11. Parkes, D.C.: Iterative Combinatorial Auctions. In: Cramton, P., Shoham, Y., Steinberg, R. (eds.) *Combinatorial Auctions*, ch. 2. MIT Press, Cambridge (2001)
12. Courcoubetis, C., Weber, R.: *Pricing Communication Networks: Economics, Technology and Modelling*, ch. 10. Wiley, Chichester (2003)
13. Mas-Colell, A., Whinston, M., Green, J.: *Microeconomic Theory*, ch. 23. Oxford University Press, New York (1995)

On Using Digital Speech Processing Techniques for Synchronization in Heterogeneous Teleconferencing

Hsiao-Pu Lin¹ and Hung-Yun Hsieh^{1,2}

¹ Graduate Institute of Communication Engineering

² Department of Electrical Engineering,

National Taiwan University,

Taipei, Taiwan, 106

hyhsieh@cc.ee.ntu.edu.tw

Abstract. As the popularity of multi-functional communication devices grows, traditional audio conferencing now may involve heterogeneous teleconferencing devices, including POTS phone, VoIP phones, dual-mode smart phones, and so on. During a multi-party audio conference involving heterogeneous devices, it is possible that a video conference is held concurrently involving a subset of devices capable of processing video streams for better the conferencing experience. In such a scenario, the need for synchronization between circuit-switched audio streams and packet-switched video streams arises. While the problem of audio-video synchronization has been extensively investigated in related work, existing solutions are limited to synchronization in packet-data networks and hence are not applicable in the target environment. In this work, we consider the problem of supporting such an overlay video conference among dual-mode phones. We first transform the audio-video synchronization problem into the problem of synchronizing circuit-switched and packet-switched audio streams. We then propose an end-to-end solution for audio synchronization that is transparent to the heterogeneous network protocol suites involved. We investigate synchronization algorithms based on digital speech processing using different acoustic features of the speech signal in the waveform, cepstrum, and spectrum domains. We evaluate the effectiveness of different algorithms under various impairments including codec distortion, line noises, packet losses, and overlapping utterances. Evaluation results show a promising direction for using DSP-based algorithms to address the synchronization problem across heterogeneous telephony systems.

Keywords: Overlay video conference, heterogeneous telephony device, dual-mode phone, VoIP.

1 Introduction

As modern communication technology advances, more and more devices have been made available for use with different telephony services, including POTS

phones, 2G/3G mobile phones, GSM/WiFi dual-mode smart phones, and even VoIP phones. A multi-party teleconference thus may involve conferees using such various types of telephony devices. Due to the disparity of device capability, however, it is possible that only a POTS audio conference can be held among such heterogeneous teleconferencing devices, despite the fact that the conferees with smart phones and pocket PCs may be capable of participating in a video conference.

To provide a better conferencing experiencing among capable devices while maintaining the audio conference, one feasible scenario is to hold the video conference atop the multi-party audio conference. Since the audio conference is held through the PSTN, a concurrent video conference based on IP thus is possible among IP-based devices such as dual-mode phones. In this way, while the audio conference involving participants with legacy POTS phones proceeds as usual, participants with dual-mode phones may still be able to leverage their hardware capability for face-to-face communications.

An important feature of such heterogeneous teleconferencing is that *the audio conference is held through the PSTN network while the video conference is held through the IP network*. Since audio and video conferences are held over different networks, heterogeneity in network environment may lead to different delays and jitters of the multimedia streams [1]. Audio and video streams thus are very likely to be asynchronous at the receiving side, potentially resulting in a perceptually unpleasant conferencing experience.

While synchronization of audio and video streams is a well-attended problem in the literature [2, 3, 4, 5, 6], conventional synchronization control schemes typically rely on the use of the common timestamp information on the audio and video streams for inter-stream synchronization, including adaptive buffer control and playout scheduling. Clearly, these schemes are proposed for operating in the IP network where it is possible to manipulate the packet header (e.g. injecting time-stamp information). They therefore cannot be used directly in the target scenario involving the circuit-switched PSTN telephony system with a very different suite of network protocols from the packet-switched IP telephony system.

To address the problem of synchronization across heterogeneous telephony systems, we therefore investigate solutions based on digital speech processing (DSP). The goal is to *avoid reliance on network protocols of the circuit-switched telephony system for providing synchronization tips*. Instead, we aim to explore the acoustic features inherent in the audio streams for synchronization at the receiving end. Acoustic features have the nice property that they can potentially prevail against different switching technologies as long as the audio streams are generated from the same utterance of the speaker. We investigate three DSP-based algorithms using acoustic features in the waveform, cepstrum, and spectrum domains. The performance of the three algorithms is evaluated against different sources of impairments including codec distortion, line noise, packet loss, and overlapping utterances in individual telephony systems. Evaluation

results show that utilizing digital speech processing techniques for synchronizing audio streams across heterogeneous telephony systems is promising.

The rest of this paper is organized as follows. Section 2 describes in details the target scenario, and how the problem of audio-video synchronization can be simplified into a synchronization problem across PSTN and IP audio streams. Sections 3, 4, and 5 present synchronization algorithms and their performance under different impairments in the waveform, cepstrum, and spectrum domains respectively. Finally, Section 6 compares the overall performance of the three algorithms and concludes the paper.

2 Synchronization in Heterogeneous Teleconferencing

In this section, we first describe the synchronization framework for supporting heterogeneous teleconferencing. We then discuss the challenges of achieving synchronization in the proposed framework.

2.1 Synchronization Framework

The scenario for overlay video conferencing atop multi-party audio conferencing is shown in Fig. 1. The audio conference is held by the audio conference server located in the PSTN network. Conferees attend this audio conference using various kinds of teleconferencing devices, including legacy POTS phones, GSM phones, GSM/WiFi dual-mode smart phones, and even laptops as IP soft phones. The video conference, on the other hand, is held among devices capable of accessing the IP network and processing video streams such as dual-mode pocket PCs and smart phones.

In such heterogeneous teleconferencing, it is necessary on dual-mode phones that the PSTN-based audio stream is synchronized with the IP-based video stream. While related work has proposed the concept of lip synchronization [7, 8] for audio-video synchronization, sophisticated processing algorithms such as face localization, lip modeling and tracking, and identification are required. We instead seek a solution without the need for processing the video stream in this paper.

Inspired by related work [6], we consider an approach that embeds audio information in the video stream for synchronization between PSTN audio and IP video streams. At the transmitter, each video frame is sent through the IP network to the receiver while carrying with it audio information (audio hash) of the current audio frame to be used by the synchronization algorithm. At the receiver, after the video stream is received from the IP network, the embedded audio hash can be extracted and used to determine the timing relationship of the IP video stream with the PSTN audio stream through audio synchronization. The problem of audio-video synchronization thus becomes the problem of synchronizing PSTN and IP audio streams.

To detail, when the dual-mode phone receives the IP video stream and PSTN audio stream, both streams are buffered for the purpose of synchronization and

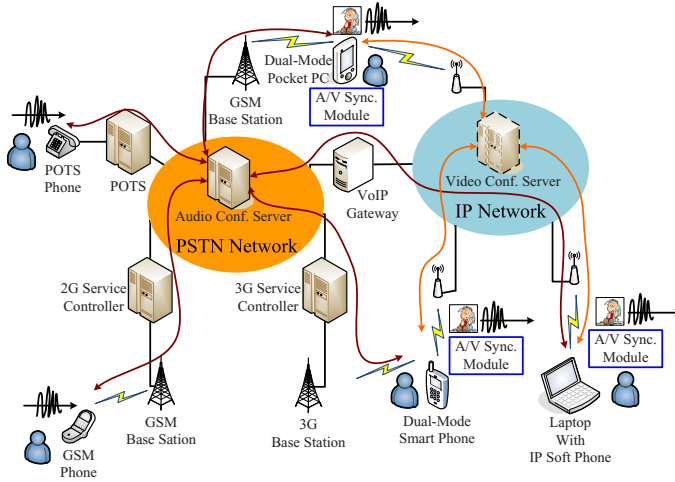


Fig. 1. Heterogeneous teleconferencing

playback. If synchronization between the two streams is necessary, the synchronization module at the receiver extracts the audio hash from the IP video stream and determines the timing relationship between the two audio streams. The resulting timing relationship is fed to the playback buffer where common synchronization control schemes can be applied. Note that whether the received audio and video streams need to be synchronized depends on the synchronization trigger that can be determined on a periodic or dynamic basis. Clearly, since *synchronization does not need to be performed for every video frame* but only when the dynamics of the two telephony systems change significantly, the requirement for real-time computation of the synchronization algorithm becomes less strict.

2.2 Challenges

To better support heterogeneous teleconferencing, it is important to synchronize circuit-switched and packet-switched audio streams with little or no reliance on circuit-switched network protocols for providing synchronization tips. Synchronization based on the inherent acoustic features of the concerned speech signal thus is one possible solution for achieving the goal. There are however challenges in such DSP-based approaches as we describe in the following:

Codec Distortion. To reduce bandwidth requirement, the audio stream is typically encoded before transmission. Different telephony services have been using different voice codecs such as the AMR (Adaptive Multi-Rate) codec in GSM and the G.723 or G.729 codec in VoIP. These voice codecs however result in a lossy compression and introduce different degrees of distortions to the original speech signal.

Packet Loss. In packet-switched networks, packets are subject to errors, delay and reordering. Such impairments may result in losses of audio frames in the packet-switched audio stream. While loss concealment algorithms have been used in VoIP to combat such impairments, nonetheless the speech signal is distorted, especially in wireless networks with bursty losses.

Line Noise. Similar to the transmission problem in IP telephony, circuit-switched audio streams may suffer from different types of noises incurred by the transmission line or user device. Such noise clearly impairs the speech signal and complicates the synchronization process.

Overlapping Utterances. In multi-party teleconferencing, the speech signal to be synchronized is “buried” inside a mixture of multiple speech signals uttered by other speakers (conferees). Different from the static, broadband noise incurred by the transmission line, interference incurred by overlapping utterances is more difficult to separate. The distortion thus incurred will impose great challenges on the synchronization algorithm.

Therefore, while a DSP-based algorithm operates directly on the speech signals and can allow for better transparency over voice switching technologies, it needs to be resilient to various distortions incurred by heterogeneous telephony systems on the speech signals. We present in the following three DSP-based algorithms for synchronization of circuit-switched and packet-switched audio streams.

3 Waveform-Based Synchronization

To determine the relative timing of two audio streams, the simplest way is to match the waveforms of the two speech signals after decoding the audio streams based on time-domain processing.

3.1 Basics of Cross Correlation (XCOR)

Cross correlation is widely used in many areas such as pattern recognition. It tries to capture how similar or different a test signal is from the specific signal. The commonly used similarity measure is the correlation coefficient, r , defined as

$$r = \frac{\sum_i^N (x(i) - m_x)(y(i) - m_y)}{\sqrt{\sum_i^N (x(i) - m_x)^2 \times \sum_i^N (y(i) - m_y)^2}}, \quad (1)$$

where $x(i)$ and $y(i)$ are the comparing signals and m_x and m_y are individual means. Therefore, if two comparing speech signals have similar waveforms, their correlation coefficient may reach a value close to 1 when the two signals are “aligned” in time. It thus can be used as a metric for determining the relative timing of two speech signals.

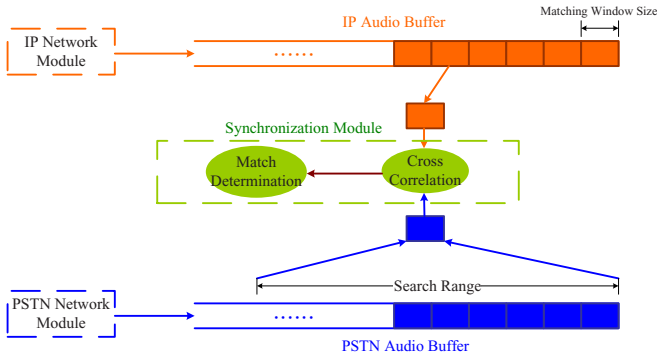


Fig. 2. XCOR synchronization module

3.2 XCOR Synchronization Module

Based on the cross-correlation function, we can design a synchronization module as illustrated in Fig. 2. Audio streams received from either circuit-switched (PSTN) or packet-switched (IP) networks are first stored in individual buffers. When the synchronization process begins, a window of speech samples (matching window) is selected from one buffer to iteratively match against a time-shifted window of speech samples of the same size in the second buffer. To reduce the computation complexity, the time shift can be limited to a *search range* if the maximum or approximate time offset of the two audio streams can be estimated beforehand. The correlation coefficient thus computed at each iteration is recorded against the time-shift value. At the end of iterations, the time shift that corresponds to the largest correlation coefficient is used as the relative time offset of the two audio streams.

3.3 Performance Evaluation

We evaluate the performance of the synchronization algorithm based on cross correlation for various types of impairments as mentioned in Section 2.2. Due to lack of space, however, we only present and discuss a subset of the results in this section.

Line Noise. The thermal noise is the most common source of noise so we focus our discussion on the distortion by thermal noise. We model the noise as the Additive White Gaussian Noise (AWGN), where the variance (σ^2) determines the energy level of noise. A noise with energy level 0.01 in our experiments corresponds to almost the lower-volume part in the source speech. Therefore, the effect of noise at this level is comprehensible. Since the noise reduces the correlation coefficient at the time shift of 0, the accuracy of correlation-based synchronization might also be affected. Fig. 3 shows the accuracy of synchronization when one of the source signal is impaired by noise. We encode one source

signal using the Adaptive Multi-Rate (AMR) codec (GSM audio stream), and the other signal using the G.729 codec (VoIP audio stream). We can observe that the accuracy of synchronization is lowered by the noise. Increasing the size of the matching window can potentially improve the accuracy, although at the cost of increased computation complexity.

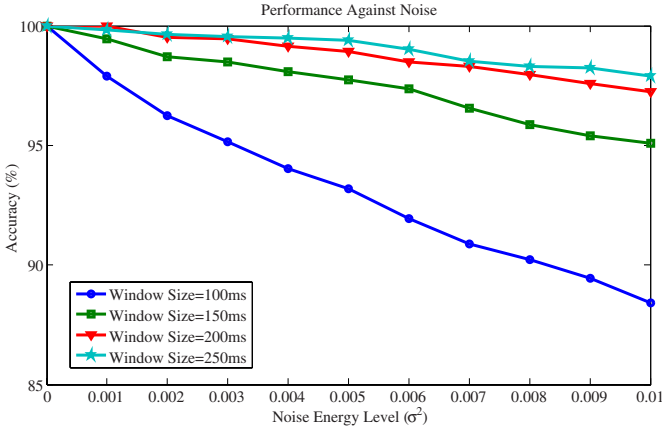


Fig. 3. Performance of XCOR against line noise

Overlapping Utterances. In a multi-party conference, it is possible that multiple speakers speak at the same time. Hence, the receiver may receive an audio stream with mixing utterances from different speakers. Fig. 4 thus shows the effect of overlapping utterances on correlation-based synchronization. From the top two sub-figures, we can observe that the correlation coefficient is substantially lowered at the point of 0 time-shift as the number of interfering utterances increases. For a small matching window such as 100 ms, the algorithm is more vulnerable to interfering utterances, and thus achieves lower accuracy compared to the case with a large matching window.

Combined Impairments. We combine all sources of impairments and evaluate the performance of correlation-based synchronization as shown in Fig. 5. We can observe that the performance is severely degraded as distortions add up. This is because cross correlation considers only the time-domain waveform, and different types of distortions change the waveform-similarity of the concerned speech signal in different ways. Even though the distortion level of individual impairments is low, overall the performance is still significantly affected. In addition, although increasing the matching window size can improve the accuracy, the performance improvement is limited considering the increase in computation complexity. Therefore, we can observe that cross correlation can sustain minor distortions on the waveform, but *as the distortion level increases or multiple distortions add up, its performance quickly drops.*

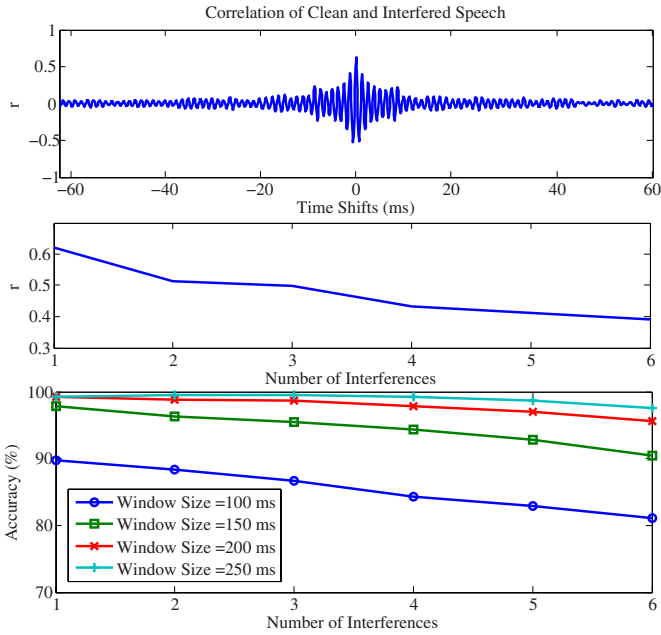


Fig. 4. Performance of XCOR against overlapping utterances

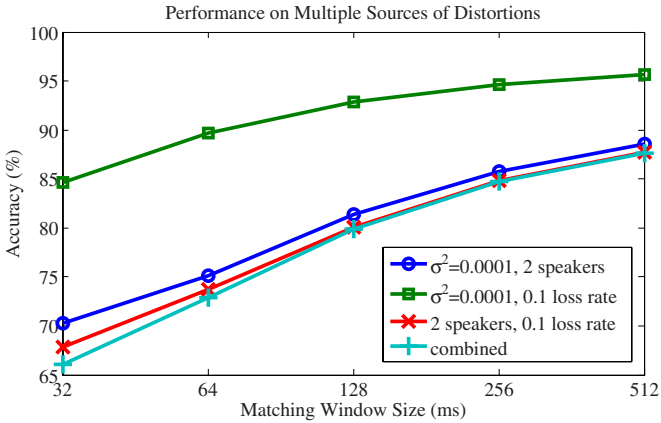


Fig. 5. Performance of XCOR against combined impairments

In conclusion, cross correlation is vulnerable to practical distortions because it considers only the time-domain waveform of the speech signal. Since the waveform is easily corrupted by distortions, a larger window size of speech samples should be used for synchronization, thus increasing computation complexity of the algorithm.

4 Cepstrum-Based Synchronization

The cepstrum of a signal is commonly used in digital speech processing applications such as voice identification and pitch detection. We thus investigate the use of cepstrum-based analysis for voice synchronization in the section.

4.1 Basics of MFCC

The cepstrum of a signal is obtained as the Fourier transform of the logarithm of the power spectrum of the signal (“spectrum of a spectrum”). It has the nice property that the *convolution* of two signals in the time domain is equivalent to the *addition* of their ceptra in the cepstrum domain. To capture the perception of the human auditory system to the speech signal, the Mel-Frequency Cepstral Coefficient (MFCC) is often used in cepstral analysis. The frequency warping introduced in the mel-scale transformation of the cepstrum allows a better representation of sound. Conventionally, for each window of speech samples (analysis window), a set of coefficients (MFCCs) is obtained as a column vector, and the index of each coefficient is referred to as the MFCC bin.

4.2 MFCC Synchronization Module

To use MFCC for audio synchronization, we first define the similarity metric for each MFCC bin similar to [9] as follows:

$$B(m, u) = \begin{cases} |m - u|, & \text{if } m + \epsilon \geq u; \\ p, & \text{otherwise,} \end{cases} \quad (2)$$

where m and u are the coefficients of the two comparing speech signals (referred to as mixed and unmixed signals) in the concerned MFCC bin respectively, ϵ is the error factor, and p is the penalty value. The penalty value is designed to differentiate the correct match at the time shift of 0 from other shifts. The error factor, on the other hand, is included for tolerance of minor faults such as those introduced by random noises. As shown in Fig. 6, different time shifts of one signal result in different signals to be compared against the other signal. For each

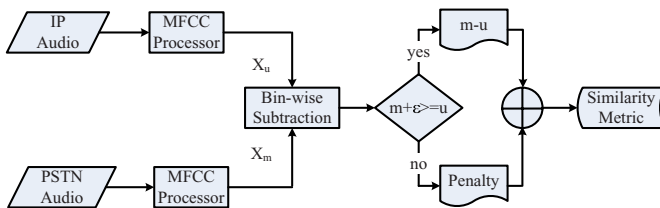


Fig. 6. MFCC synchronization module

comparing signal, a window of speech samples (matching window) consisting of multiple analysis windows is transformed into multiple columns of MFCCs. $B(m, u)$ is then computed for every MFCC bin in one column, and values over all MFCC bins over all columns are summed up as the similarity metric of the two comparing signals. Once the similarity metric for the two comparing signals (with different time shifts) is determined, the synchronization process can proceed.

4.3 Performance Evaluation

To evaluate the performance of the synchronization algorithm based on MFCC, we set the size of the analysis window to 32ms (corresponding to one matching column). The size of the matching window is varied depending on the number of analysis windows (matching columns) included for calculating the similarity of the two signals. Due to lack of space, we present and discuss only a subset of the results in the following.

Packet Loss. When packet losses occur in VoIP, conventional approaches apply packet loss concealment methods where the missing frame is filled using the previously received frame. Fig. 7 reveals the robustness of MFCC against packet losses when the lost frame is concealed by duplicating the frame in the previously received packet. Although a small number of matching columns might not be sufficient to differentiate the correct match from other shifts, the case with 4 matching columns performs reasonably well for a packet loss rate of less than 20%.

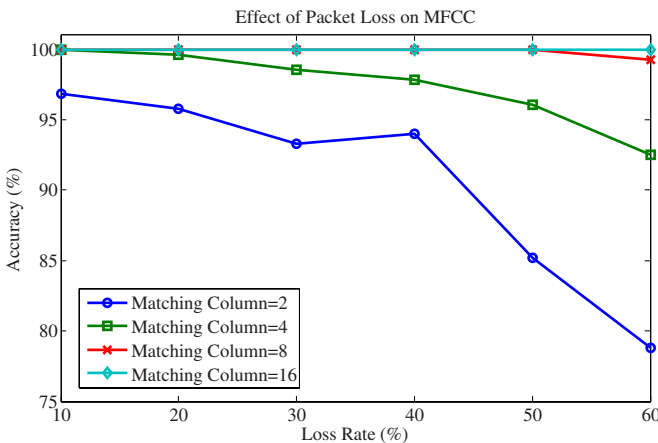


Fig. 7. Performance of MFCC against packet loss

Overlapping Utterances. As shown in Fig. 8, once the PSTN audio contains interfering utterances from other speakers, the percentage of accuracy drops significantly. Even for the case with 16 matching columns (about 512ms of matching window), the performance of the MFCC-based algorithm drops from 100% to about 60% as the number of utterances in the PSTN audio stream increases from 1 to 3. This is because as the number of speakers increases, the coefficients in each MFCC bin are heavily distorted and it becomes more difficult to differentiate the correct match from other shifts using the penalty value. Therefore, the MFCC-based synchronization algorithm also shows the problem of vulnerability to interfering utterances.

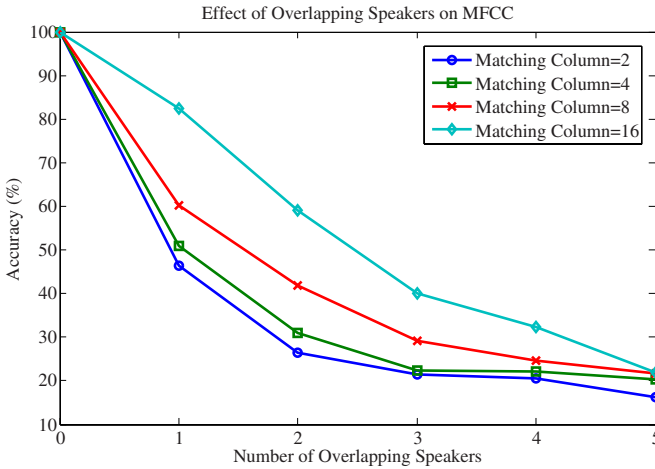


Fig. 8. Performance of MFCC against overlapping utterances

Combined Impairments. Fig. 9 shows the performance of MFCC when multiple sources of distortions occur. We can observe that since MFCC is vulnerable to overlapping utterances, whenever an audio stream involves mixed utterances, its performance degrades significantly. For the case with no overlapping speakers, however, MFCC performs reasonably well under additional sources of distortion if the size of the matching window is sufficient (about 256ms).

To sum up, as long as the size of the matching window is appropriately chosen, using the similarity of MFCC bins for synchronization seems to be a viable option since it is robust against many types of waveform distortions. However, its performance is severely degraded against overlapping utterances in the PSTN audio stream. In addition, using a larger size of the matching window is not effective in this case. This is a potential problem with MFCC since in a practical conference, especial during a keen discussion, many speakers may speak at the same time, and thus the PSTN audio stream is likely to be a mixture of utterances from multiple speakers.

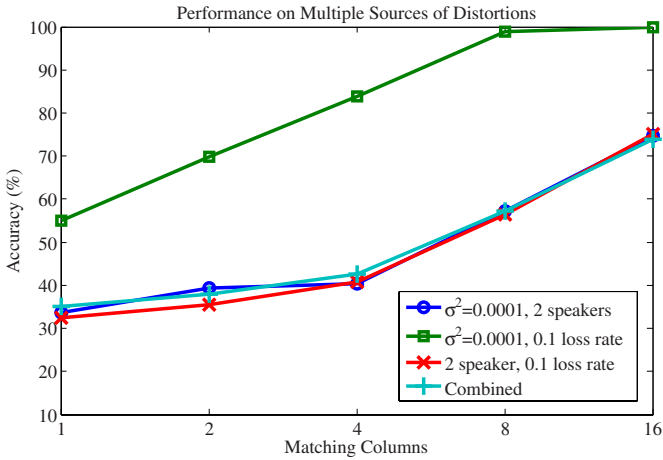


Fig. 9. Performance of MFCC against combined impairments

5 Spectrum-Based Synchronization

Waveform-based synchronization exploits the similarity of audio streams in the time domain. Spectrum-based synchronization, on the other hand, exploits the similarity in the frequency domain.

5.1 Basis of Spectrogram (SPGM)

Since the human speech varies with time, a direct transform of the speech signal into the frequency domain will lose many useful information. Instead, a short-time Fourier transform (STFT) is applied on the speech signal on a window-by-window (time frame) basis. The squared magnitude of the STFT over time thus obtained is the spectrogram of the signal, and it shows the variation of the spectral density of the signal over time.

While the frequency-domain representation of a signal ideally is equivalent to the time-domain representation, frequency-domain processing has several nice properties. An important property of the spectrogram that we explore in this paper is the property of *window-disjoint orthogonality (W-DO)* for human speech [10, 11]. It has been discovered that a speech signal is sparsely distributed in its time-frequency representation and, as a result, different speech utterances (except speech babbles) tend not to overlap (orthogonal) in the time-frequency plot such as the spectrogram. With the assumption of W-DO, the value of each frequency bin at a certain time frame can be considered as being contributed by a single speaker as we show in Fig. 10. Many blind speech separation techniques based on the approximate W-DO of speech have been proposed [12, 13]. The sparsity in spectrogram provides a helpful tool for voice synchronization *if the speech signal to be synchronized is “buried” inside a mixture of multiple speech signals.*

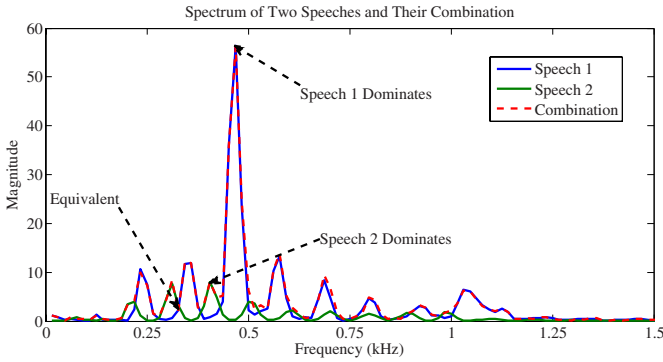


Fig. 10. Spectrum of mixture is usually dominated by one utterance

5.2 SPGM Synchronization Module

To explain the operation of the synchronization algorithm based on spectrogram, assume that the PSTN audio stream is mixed with multiple utterances, and the goal is to synchronize the unmixed IP audio stream against the mixed PSTN audio stream. Based on the concept of W-DO, some frequency bins in the spectrogram of the PSTN speech signal are dominated by the utterance of the concerned speaker, and hence they can be used for similarity measure against the IP audio stream. We can choose only those “significant” frequency bins for minimizing the disturbance of overlapping utterances in the synchronization process.

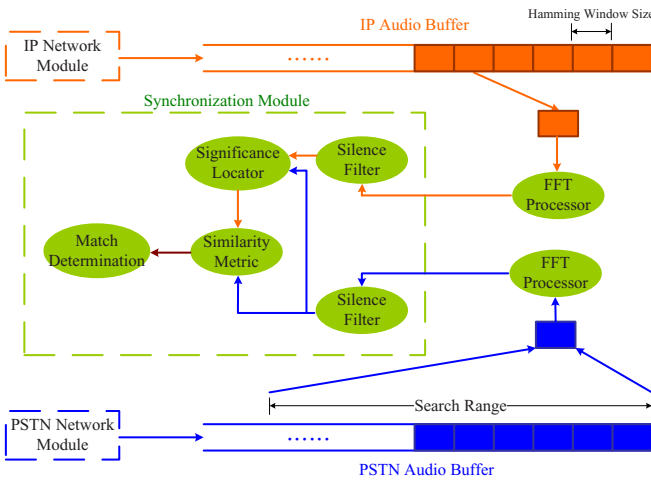


Fig. 11. SPGM synchronization module

The synchronization module based on spectrogram is shown in Fig. 11. When the synchronization process begins, a window of speech samples (unmixed signal) is selected from the IP audio buffer and sent to the *FFT processor* to compute the spectrogram. The speech samples of the PSTN audio buffer within the search range (of time shifts) is also sent to the FFT processor for generating the second spectrogram. The *silence filter* inside the synchronization module is used to exclude silence frames to avoid unnecessary and error-prone matching. For each possible time shift of the two spectrograms, the *significance locator* infers from the two spectrograms and determines which frequency bins should be included for calculating the similarity. Let $m(\tau, \omega)$ and $u(\tau, \omega)$ be the spectrogram values of the mixed and unmixed signals at time τ and frequency bin ω respectively. The metric $S(\tau, \omega) = \left| \frac{m-u}{u} \right|$ is then used for determining the significance of bin ω of frame τ . The similarity (dissimilarity) metric of the two comparing signals is calculated by summing the significance of all bins with $S(\tau, \omega) < \eta$ over all time frames (matching columns). The time shift that yields the lowest value is determined as the relative time offset of the two audio streams.

5.3 Performance Evaluation

To evaluate the performance of the synchronization algorithm based on spectrogram, we set the analysis window to 64ms with 32ms overlap. Each matching column thus represents a 32ms time-shift from adjacent columns.

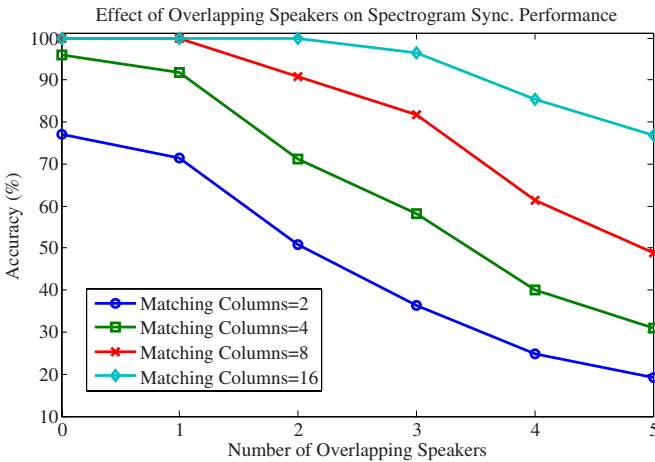


Fig. 12. Performance of SPMG against overlapping utterances

Overlapping Utterances. From Fig. 12 we can observe that the percentage of accuracy still drops as the number of interfering speakers increases. This is because the sparsity is somewhat compromised as the number of speech utterances

increases. However, comparing to the performance of MFCC-based algorithm, this spectrogram-based algorithm can achieve better accuracy. If we consider the case with two interfering speakers, the spectrogram-based algorithm can achieve more than 90% of accuracy if 8 or more matching columns are applied. The MFCC-based algorithm, on the other hand, achieves only 40% of accuracy as shown in Fig. 8.

Combined Impairments. Fig. 13 shows the performance of the spectrogram-based algorithm when multiple sources of distortions occur. The spectrogram-based algorithm shows performance benefits compared to the other two algorithms when the number of speakers increases. In addition, its performance is consistently improved as the number of matching columns increases.

In conclusion, for the case of overlapping utterances, the synchronization algorithm based on spectrogram achieves better performance compared to the other two algorithms due to speech sparsity. As for other impairments such as packet losses, however, the spectrogram-based algorithm does not show significant performance benefits over the other two.

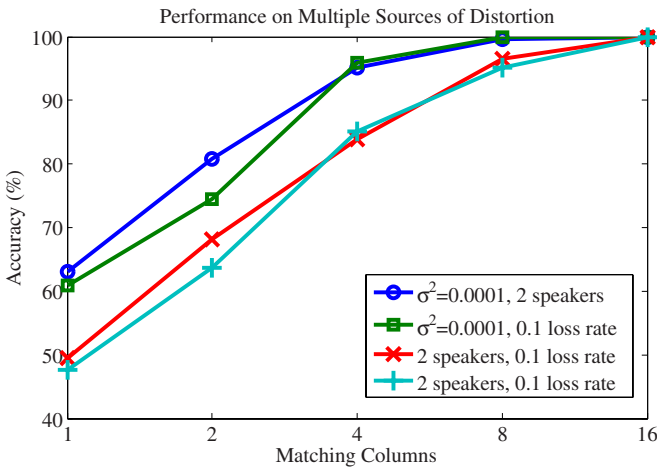


Fig. 13. Performance of SPGM against combined impairments

6 Conclusions

To compare the performance of the three algorithms, we show in Fig. 14 their performance under all sources of distortions as described in Section 2.2. The two audio streams are encoded using AMR and G.729 respectively, the packet loss rate is set to 10%, the variance of the noise is set to 10^{-4} , and the number of overlapping utterances is set to 2 and 4 for comparison. We can observe that the spectrogram-based algorithm outperforms the other two. The XCOR-based synchronization algorithm is limited in its performance by multiple sources

of distortions. As for the MFCC-based synchronization algorithm, since it is vulnerable to overlapping utterances, the performance is the worst among these three. However, without overlapping utterances, the MFCC-based algorithm can typically achieve better performance than the spectrogram-based algorithm. A hybrid algorithm combining multiple metrics may potentially be used for audio synchronization in future work.

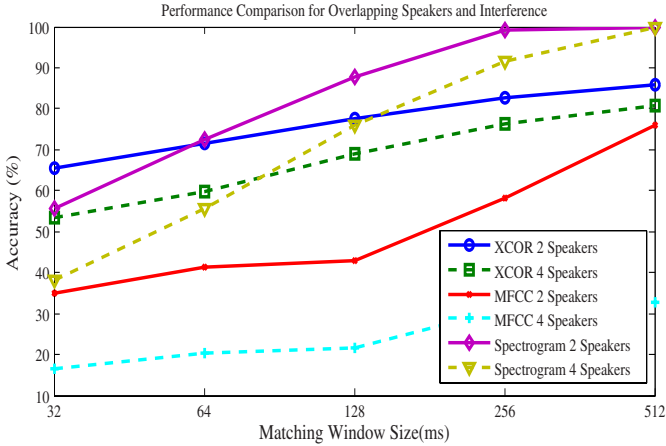


Fig. 14. Performance of the three algorithms against combined impairments

In conclusion, we have considered in this paper the problem of supporting video conferencing atop a multi-party audio conference with heterogeneous telephony devices. We have identified the importance of synchronizing the IP video stream and the PSTN audio stream in the target scenario. We have transformed the audio-video synchronization problem into the problem of synchronizing audio streams across heterogeneous telephony systems. To address the synchronization problem between circuit-switched and packet-switched audio streams, we have proposed an end-to-end solution framework transparent to the heterogeneous network protocol suites involved. Under this framework, we have investigated three synchronization algorithms based on digital speech processing in the waveform, cepstrum, and spectrum domains. Evaluation results show that such DSP-based techniques are an appealing solution toward addressing the target problem.

Acknowledgment

This work was supported in part by funds from the Excellent Research Projects of the National Taiwan University under Grant 97R0062-06.

References

1. Hsieh, H.-Y., Li, C.-W., Lin, H.-P.: Handoff with DSP support: Enabling seamless voice communications across heterogeneous telephony systems on dual-mode mobile devices. *IEEE Transactions on Mobile Computing* 8(1), 93–108 (2009)
2. Liu, C., Xie, Y., Lee, M.J.: Multipoint multimedia teleconference system with adaptive synchronization. *IEEE Journal on Selected Areas in Communications (J-SAC)* 14, 1422–1435 (1996)
3. Xie, Y., Liu, C., Lee, M.J., Saadawi, Y.N.: Adaptive multimedia synchronization in a teleconference system. *ACM/Springer Multimedia Systems* 7(4), 326–337 (1999)
4. Kim, C., Seo, K.-D., Sung, W., Jung, S.-H.: Efficient audio/video synchronization method for video telephony system in consumer cellular phones. In: *Proceedings of the ICCE 2006 Consumer Electronics*, January 2006, pp. 137–138 (2006)
5. Liu, H., Zarki, M.E.: A synchronization control scheme for real-time streaming multimedia applications. In: *Proceedings of 13th Packet Video Workshop* (April 2003)
6. Yang, M., Bourbakis, N., Chen, Z., Trifas, M.: An efficient audio-video synchronization methodology. In: *Proceedings of the IEEE International Conference on Multimedia and Expo.*, July 2007, pp. 767–770 (2007)
7. Lie, W.-N., Hsieh, H.-C.: Lips detection by morphological image processing. In: *Proceedings of ICSP 1998*, pp. 1084–1087 (1998)
8. Zoric, G., Pandzic, I.S.: A real-time lip sync system using a genetic algorithm for automatic neural network configuration. In: *Proceedings of the IEEE International Conference on Multimedia and Expo.*, July 2005, pp. 1366–1369 (2005)
9. Cutler, R., Bridgewater, A.: Audio/video synchronization using audio hashing. Patent No. US 2006/0291478 A1 (December 2006)
10. Jourjine, A., Richard, S., Yilmaz, O.: Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2000, pp. 2985–2988 (2000)
11. Rickard, S., Yilmaz, O.: On the approximate W -Disjoint Orthogonality of speech. In: *Proceedings of ICASSP*, May 2002, pp. 13–17 (2002)
12. Yilmaz, O., Rickard, S.: Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing* 52(7), 1830–1847 (2004)
13. Shan, Z., Swary, J., Aviyente, S.: Underdetermined source separation in the time-frequency domain. In: *Proceedings of ICASSP*, September 2007, pp. 945–948 (2007)

Interference-Free Coexistence among Heterogenous Devices in the 60 GHz Band

Chun-Wei Hsu and Chun-Ting Chou

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
r96942118@ntu.edu.tw, chuntingchou@cc.ee.ntu.edu.tw

Abstract. With its abundant bandwidth and worldwide availability, the 60 GHz band has been considered as a promising solution to provide multi-Gbps wireless transmission. Different standard bodies and industrial interest groups start various projects to develop technologies in the 60 GHz band for applications such as high-definition (HD) and fast file transfer. However, until now very little efforts are made to ensure interference-free coexistence between these technologies. In this paper, we investigate the interference problem in the 60 GHz band. The ECMA-387 standard is used as a study case to illustrate how some simple techniques can mitigate the interference among heterogenous devices in the 60 GHz band. We conduct both mathematical analysis and simulations to demonstrate the performance of these simple techniques and identify some problems for future improvements.

Keywords: heterogenous, interference, coexistence, wireless PANs.

1 Introduction

In the last decade, data rates and file sizes of multimedia applications have increased substantially. These applications drive the data rate beyond the megabit-per-second (Mbps) level to a formidable gigabit-per-second (Gbps) level. Take high definition (HD) video as an example. The raw data rate of a full HD video with a resolution of 1920*1080 pixels is as high as 2.98 Gbps.¹ In order to support transfer of such high-date rate videos between consumer electronics, High Definition Multimedia Interface (HDMI) was developed. The HDMI cable can transfer uncompressed HD video within a distance of 10 meters at a rate of 4.7 Gbps.

In addition to HD videos, many other applications including bulk file transfer, system backup and computer ducking stations can benefit from multi-Gbps transmission. Although these applications generally do not require strict quality of service (QoS) as HD videos, the transfer time is always a crucial performance metric. For example, consider synchronizing an 80G-iPod with the music library in a personal computer. It takes more than 20 minutes if using a USB 2.0 cable (at a full speed of 480 Mbps). However, it takes only 3 minutes if using the latest USB 3.0 cable (up to 4.8 Gbps).

¹ The refreshing rate is 60 frames per second, and each pixel is represented by 24 bits.

Although many “wired” solutions (e.g., HDMI or USB 3.0) are developed to support multi-Gbps transmission, doing so over a wireless link is never an easy task. In principle, one can increase the transmission rate of a wireless link by increasing (1) the spectrum efficiency, and/or (2) the transmission bandwidth. By using the advanced techniques such as orthogonal frequency division multiplexing (OFDM) and multi-input multi-output (MIMO), one might be able to “squeeze” out 15 bits per second per Hertz [1]. However, even with such high spectrum efficiency, a bandwidth of 200 MHz is needed to achieve a 3-Gbps transmission rate. One can find that the real challenge to achieve multi-Gbps wireless transmission is to allocate a sufficient amount of bandwidth in the already-crowded wireless spectrum.

Lately, the 60 GHz band becomes an attractive solution to multi-Gbps wireless communications. The 60 GHz band has a common 3.5 GHz bandwidth worldwide, up to 9 GHz in Europe, and up to 7 GHz in North America and Japan. Figure 1 shows the spectrum availability in these regions. Thanks to its abundant bandwidth and worldwide availability, the 60 GHz band can provide multi-Gbps transmission with a much relaxed spectrum efficiency, which leads to a simpler hardware design. For example, one can use a bandwidth of 2 GHz in the worldwide 60 GHz band, with a spectrum efficiency of 1.5 bits per second per Hertz, to “easily” achieve a data rate of 3 Gbps.

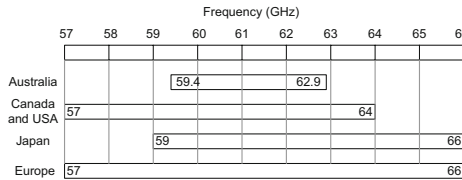


Fig. 1. Spectrum availability around 60 GHz band

1.1 Multiple Competing Standards

In views of the promising potential in the 60 GHz band, different standard bodies and interest groups have started various projects for 60-GHz communications. These projects include WirelessHD developed by the WirelessHD Consortium [2], IEEE 802.15.3c developed by the IEEE 802.15 Task Group 3c [3], WiGig developed by WiGig Alliance [4], IEEE 802.11ad developed by the IEEE 802.11 VHT Study Group [5], and finally, the ECMA-387 [7] developed by the Ecma International TC48.

WirelessHD targets on streaming uncompressed full-HD audio and video at a distance of 10 meter. WirelessHD specifies two physical (PHY) modes: High Rate PHY (HRP) and Low Rate PHY (LRP). HRP adopts the OFDM and provides a throughput of over 3 Gbps (when using 16-QAM) for video streaming. LRP also adopts the OFDM but only supports up to 40Mbps transmission for lower power consumption. The LRP is used for transmitting and receiving control messages and cannot be used when there is video streaming in vicinity.

IEEE 802.15.3c is designed for general wireless personal area networks (PANs) and supports data rates from 1 Gbps to more than 5 Gbps. IEEE 802.15.3c defines three PHY modes, including Single Carrier mode (SC PHY), High Speed Interface mode (HSI PHY), and Audio/Visual mode (AV PHY). SC PHY uses single carrier with adaptive modulations to achieve a data rate of up to 5.28 Gbps. On the other hand, HSI PHY and AV PHY (dedicated for video streaming) adopt the OFDM to provide data rates of 5.77 and 3.8 Gbps, respectively.

IEEE 802.11ad is designed for wireless local area networks (LANs) and aims to support at least 1 Gbps transmission at a range of at least 10 meters. As a member of the 802.11 family, IEEE 802.11ad relies on a so-called Fast Session Transfer to provide “*a seamless transfer of an active session from the 60 GHz band to the 2.4/5 GHz band, and vice versa.*” This function allows devices to switch among different PHYs for either higher transmission rates (in 60 GHz band), or extended ranges and better reliability (in 2.4/5 GHz band) depending on the network environment.

The ECMA-387 standard supports multi-Gbps wireless transport for both bulk data transfer and multimedia streaming. In order to support these applications, the ECMA-387 standard defines three device types: Types A, B, and C, each with a different level of capability and hardware complexity. According to ECMA-387, Type A devices support both single carrier block transmission (SCBT) and OFDM to provide more than 6 Gbps transmission at 10 meters for full-HD video streaming and Wireless PAN applications. Type B devices use a simpler single carrier modulation to support a lower data rate (3.175 Gbps) at 3 meters. Type C devices use an even simpler single carrier modulation (OOK and 4-ASK) to support point-to-point data exchange without QoS guarantees. Type C devices only communicate within a range less than 1 meter using a data rate of up to 3.2 Gbps.

1.2 Challenges – Interference among Heterogenous Devices

As one can see from the above discussion, different solutions have been developed to enable a wide range of applications in the 60 GHz band. On one hand, we can find the best solution for each application. On the other hand, devices based on different solutions will be collocated in the same 60 GHz band. Since these solutions adopt different PHY-layer technologies, network architectures and channelization, the collocated devices inevitably will interfere with each other. This problem is referred to as interference among heterogenous devices in this paper.

Interference among heterogenous devices is not a unique problem in the 60 GHz band. Take 2.4 GHz ISM band as an example. The WiFi, Bluetooth and ZigBee devices can interfere with each other when using overlapping frequency bands. However, interference in the 60-GHz band could be a much serious problem compared to that in 2.4 GHz band for the following reasons. First, the number of channels in the 60 GHz band is much less than in the 2.4 GHz band. In the 60 GHz band, the channel bandwidth is generally between 1 to 2 GHz in order to support multi-Gbps transmission. As a result, only 3 or less channels

can be defined. In the 2.4 GHz band, devices usually have more than 10 channels to operate (e.g., 11 channels for WiFi, or 16 channels for ZigBee devices). With less channels available, devices in the 60 GHz band have more difficulties to avoid interference from other heterogenous devices.

Second, the maximum allowed power in the 60 GHz band is much higher than in other unlicensed bands. The Equivalent Isotropically Radiated Power (EIRP) in the 60 GHz band can be as high as 40dbm. With such a transmission power, along with the use of directional antennas, the interference among heterogenous devices in the 60 GHz band simply cannot be ignored as it is in the 2.4 GHz band.

Finally, both HD video streaming and bulk data transfer are high-duty cycle applications. Even with the multi-Gbps transmission rates, the wireless link can easily be fully-loaded by a single high-duty cycle application. In order to reduce overhead and fully utilize the link capacity, most of the existing 60 GHz solutions adopt the master-slave architecture with a reservation-based channel access. Unlike the widely-adopted, contention-based channel access in the 2.4 GHz band (e.g., carrier sense multiple access used by WiFi or ZigBee devices), devices using the reservation-based channel access do not perform energy detection before accessing a channel. As a result, heterogenous devices in the 60 GHz band do not have the chance to avoid each other in time domain as WiFi and ZigBee devices.

In this paper, we investigate the interference among heterogeneous device in the 60 GHz band. Instead of harmonizing the physical-layer signal formats, we approach this problem from the perspective of the medium access control (MAC) layer. We will use the ECMA-387 standard as our study case and show how heterogenous devices may be able to coexist with each other without a common PHY layer.

The rest of this paper is organized as follows. In Section 2, we give an introduction to the mechanisms for interoperability and coexistence in the ECMA 387 standard. In Section 3, we examine different interference scenarios, and identify some potential problems when using the ECMA-387 solutions. In Section 4, we analyze and simulate the identified problems using OPNET Modeler. Finally, we conclude our work in Section 5.

2 Case Study: ECMA-387

The ECMA-387 standard is an all-purpose standard that supports a wide range of applications in the 60 GHz band. To accommodate various applications while taking into account device complexity, the ECMA-387 standard defines three device types, each for some specific applications. The devices of different types use different PHY modes and network architectures, and thus, treat each other as heterogenous devices. To address this issue, several mechanisms are specified in the ECMA-387 standard for inter-operation and coexistence among these devices. In this section, we give a detailed description of these mechanisms and explain the design rationale.

2.1 Heterogeneous Devices: Three Device Types

ECMA-387 defines three types of devices, namely the advanced Type A device, the second simple Type B device, and the simplest Type C device. The target applications of Type A devices include HD video streaming and general applications in a Wireless PAN. The target applications of Type B devices are point-to-point, short-range (1-3 meters) video streaming and data transfer. Type C devices target the point-to-point fast file download in less than 1 meter. Among these three types of devices, Type A devices are considered as the “high end, high performance” device; Type B devices are considered as “economic” devices, and the Type C devices are considered as the “low-end” cheap devices.

Each type of devices uses a basic PHY mode specifically designed for that type. These basic PHY modes are mode-A0, mode-B0, and mode-C0 for Type A, Type B, and Type C devices, respectively. The mode-A0 PHY uses SCBT with BPSK and provides 0.397 Gbps within 10 meter transmission ranges. The mode-B0 uses SC with DBPSK and provides 0.794 Gbps within 1 to 3 meter transmission ranges. The mode-C0 uses SC with OOK and provides 0.8 within less than 1 meter ranges. If a pair of communicating devices belong to the same device type, they communicate using the PHY mode of the device type.

In addition to supporting its own basic PHY mode, a device is also required to support the basic PHY modes of less advanced devices. For example, a Type A device is required to support both mode-B0 and mode-C0 PHYs while a Type B device is required to support the mode-C0 PHY. These additional requirements facilitate the inter-operation among heterogeneous devices without incurring too much PHY complexity to the more advanced Type A and Type B devices. For example, a Type A device can then communicate with Type B devices using the mode-B0 PHY, and communicate with Type C devices using the mode-C0 PHY. Similarly, a Type B device can communicate with Type C devices using the mode-C0 PHY.

Based on the above designs in the ECMA-387 standard, the mode-C0 PHY seems as a common PHY for all devices. With this common PHY, devices of different types may exchange information with each other to avoid interference. Unfortunately, the transmission range of the simple mode-C0 PHY is only around 1 meter. Therefore, devices that are beyond this range cannot communicate using the mode-C0 PHY and may still interfere with each other. As we will explain later, the ECMA-387 relies on a dual-beacon protocol to address the interference among homogeneous devices.

2.2 Interoperability

In the ECMA-387 standard, interoperability is referred to as data communication between heterogeneous devices. To enable inter-operation, devices of different types form a Master-Slave Pair (MSPr). In the MSPr, the more advanced device is the master device while the simpler device is the slave device. For instance, in an MSPr established by a Type A device and a Type B device, the Type A device is the master and the Type B device is the slave. In an MSPr established

by a Type B device and a Type C device, the Type B device is the master and the Type C is the slave.

Data transmission in an MSPr is carried out by exchange of poll frames between the master and slave devices. In the poll frames, the master device specifies the time for the slave device to transmit/receive the data as well as the time to receive the next poll frame. The slave device follows the timing provided in the poll frame and transmits/receives during the specified interval. The timing structure of an MSPr is shown in Figure 2. Since the simpler slave device does not support more advanced PHY modes, transmission of polls and data frames must use the basic PHY mode of the slave device. As a result, the transmission rate and range of an MSPr is limited by the capability of the slave device. For example, the transmission range of a Type A-C MSPr cannot exceed 1 meter. This is an unfortunate compromise due to the tradeoff between the performance and complexity.

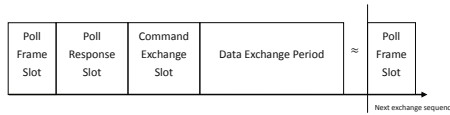


Fig. 2. Timing structure of a Master Slave Pair (MSPr)

2.3 Coexistence

Devices of different types in the ECMA-387 may interfere with each other when they are collocated in the same channel but not in the inter-operation mode. Figure 3 shows an example of how two independent pairs of Type A devices and Type B devices interfere with each other. Here, the two Type A devices use the mode-A0 PHY for their communication while the two Type B devices use the mode-B0 PHY. One can see that the Type A device X and the Type B device Y interfere with each other as their antennas point to each other. The ECMA-387 addresses the interference problems differently depending on the devices involved in the interference. For interference that involves Type A and Type B devices, a Dual-beacon Protocol is adopted. For interference that involves Type C devices, a channel sensing mechanism is adopted. These two approaches are detailed in the following two subsections.

Dual-beacon Protocol. The ECMA-387 standard requires Type A and Type B devices to send beacons periodically using their own basic PHY modes. The purposes of beacon transmission are two-fold. First, the beacons are used by the devices of the same type to synchronize with each other. Second, a device declares in the beacons its reservation of the channel time. Once receiving the reservation information, devices of the same type can avoid interfering with that device by honoring its reservation. The timing structure established by the beacons is shown in Figure 4. With this timing structure, an advanced device can

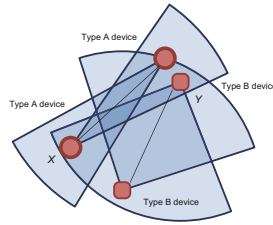


Fig. 3. Coexistence between the AA-BB communicating pairs

communicate with another advanced device of the same type while communicating with a simpler device via the MSPr. For example, as also shown in Figure 4, Type A device *X* reserves time block 1 to communicate with Type A device *Y*, and reserves time block 2 to inter-operate with Type C device *Z*. Since the beacons from devices of different types use different PHY modes, a Type-B device cannot receive and decode the mode-A0 beacons to synchronize and honor the Type A device’s reservation. On the other hand, a Type A device is able to receive and decode the mode-B0 beacons and thus may avoid interfering with the Type B devices. However, the transmission range of the mode-B0 PHY is only 3 meters. As a result, a Type A device that is more than 3 meters from a Type B device is still unaware of the Type B device’s reservation and thus cannot avoid interfering with the Type B device.

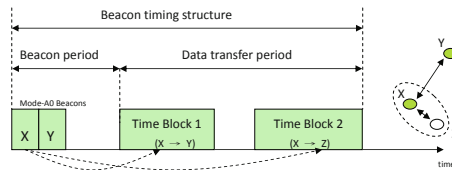


Fig. 4. Timing structure established via transmission and reception of beacons

To address the interference between the Type A and Type B devices, the ECMA-387 standard requires a Type B device to send mode-A0 beacons as well. However, the Type B device does not need to receive and decode the mode-A0 beacons. The mode-A0 beacon is transmitted solely for the Type A devices to avoid interfering with the Type B devices. Since the transmitter of a mode-A0 PHY is much simpler compared to the receiver of a mode-A0 PHY, this additional requirement does not incur too much hardware complexity to the second simple Type B device. The transmission of both mode-A0 and mode-B0 beacons by the Type B devices is referred to as Dual-beacon protocol in the ECMA-387 standard.

The timing structure of the Dual-beacon Protocol is illustrated in Figure 5. Each Type B device sends a mode-A0 beacon right after the mode-B0 beacon.

The mode-A0 beacon sent by a Type B device contains identical information as the mode-B0 beacon. With these information, a Type A device can honor a Type B device's reservation and coexist with the Type B device even they are 10 meters apart.

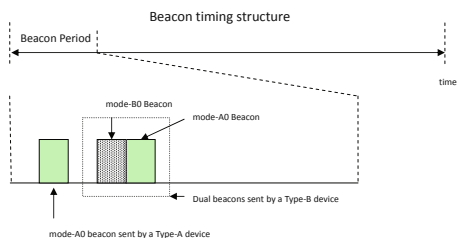


Fig. 5. Timing structure in Dual-beacon Protocol

Channel sensing with prioritized access. Type C devices are considered as the lowest-level devices in the ECMA-387 standard. In order to reduce the hardware complexity, the Type C device are not to required to transmit and receive beacons. For two Type C devices to communicate with each other, an MSPr is established with one Type C device being the master device and the other being the slave device.

Without transmission and reception of any beacon, a Type C device may interfere with the Type A and Type B devices as well as being interfered by these devices. The ECMA-387 standard addresses this issue by requiring a Type C device to transmit only when the channel is free of non-Type C transmission. To achieve that, a Type C device scans the channel for a duration that is long enough to determine that the channel is free of non-Type C transmission, before starting an MSPr. In addition, a Type C device aperiodically ceases transmission to scan for non-Type C transmission after the MSPr is established. Both the master and slave Type C devices must perform the scans to minimize the interference. When detecting non-Type C transmission, the master Type C device suspends the MSPr for an additional period of time. If the channel is free of non-type C transmission after the suspension, the master device restarts the MSPr. If a slave Type C device detects non-Type C transmission, it informs the master Type C device. The master Type C device reacts as if the non-Type C transmission is detected by itself.

The aforementioned rules make the Type C devices in the ECMA-387 standard as the lowest priority devices in terms of channel access. Given that the Type C devices targets on the low-cost application with no QoS guarantee, sacrificing the performance of the Type C devices seems not a bad option (at least from the viewpoint of the ECMA-387 standard).

Synchronization issues. The Dual-beacon protocol enables a Type A device to honor a Type B device's reservation and thus to avoid interfering with the Type B device. To achieve such interference-free coexistence, these devices must

synchronize with each other. Otherwise, the timing information in the beacons for channel reservation is useless.

Synchronization between devices using the Dual-beacon Protocol is a tricky issue. Given that the Type B devices can only transmit (not receive or decode) the mode-A0 beacons, the Type B devices are unable to use the mode-A0 beacons from the Type A devices for synchronization. The ECMA-387 standard specifies unique rules to address the synchronization problem of the Dual-beacon Protocol. These rules are listed as follows.

- 1) A Type-A device X shall synchronize to the slowest Type B device, unless the Type-B device is one of the slave devices in X 's MSPr.
- 2) A Type-A device indicates itself as a forced synchronized device in the beacons if it synchronizes to a Type B device or a forced synchronized device.
- 3) A non-forced synchronized device X shall synchronize to a forced synchronized device Y , unless Y is already forced to synchronize to X .
- 4) A forced synchronized device shall not synchronize to another forced synchronized device.
- 5) A device indicates itself as a forced synchronized device in the beacons if it synchronizes to a forced synchronized device.
- 6) A Type A device X shall synchronize to the slowest Type A device Y , if there is no Type B device or forced synchronized device other than X 's slave devices.
- 7) A slave device shall synchronize to its master device in the MSPr.
- 8) A device indicates itself as a forced synchronized device in the beacons if it is a slave device in an MSPr.
- 9) A Type B device X shall synchronize to the slowest Type B device Y , if X is not a slave device and there is no forced synchronized device other than X 's slave devices.

The forced synchronization is designed to address the case when a device detects another device but finds out that device cannot synchronize with itself (i.e., one-way communication between Type A and Type B devices). Although these rules solve most of the synchronization problems in the ECMA-387 standard, we show in the next section that there still exist some problems that need further consideration.

3 Synchronization Problems for Interoperability and Coexistence

The ECMA-387 standard defines synchronization rules for interference-free coexistence among heterogenous device. In this section, we thoroughly examine these rules in different network scenarios. We focus on the scenarios with communication pairs formed by Type A devices and Type B devices. The Type C devices are the lowest-priority devices and have little impact on the performance of Type A and Type B devices. The scenarios of our interests are further categorized into four cases, and are denoted as AA-BB, AA-AB, AB-BB, and AB-AB, respectively.

3.1 Case I: AA-BB Pairs

The logic topology of Case I is shown in Figure 6. Since the two Type B devices are not the slaves of any device and do not understand the mode-A0 beacons, they simply synchronize to each other according to Rule no.9. The two Type A devices are forced to synchronize to the Type B devices directly or indirectly, depending on whether or not receiving the mode-A0 beacons from the Type B devices. If yes, the Type A device is forced to directly synchronize to the Type B device according to Rule no.1. Otherwise, the Type A device is forced to synchronize to another forced synchronized Type-A device according to Rule no.3. Note that if none of the Type A devices is a forced synchronized device, it means that both of the Type A devices are outside the coverage of the Type B devices. Thus, the two pairs, AA and BB, do not interfere with each other and do not need to synchronize to each other.

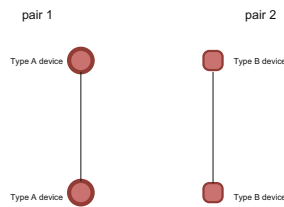


Fig. 6. Case I: AA-BB pairs

3.2 Case II: AA-AB Pairs

The logic topology of Case II is shown in Figure 7. Since the Type B device is the slave of its master Type A device, the Type B device is forced to synchronize to that Type A device according to Rule no.7. If any of the two Type A devices in the AA-pair is in the coverage of the Type B device, these two Type A devices will be forced to synchronize to the Type B device according to Rule no.3. If any of the Type A devices in the AA pair is also in the coverage of the Type A device in the AB-pair, the Type A device in the AB-pair will also be forced to synchronize to the Type A devices in the AA-pair according to Rule no.3.

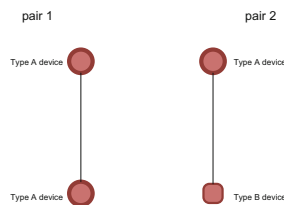


Fig. 7. Case II: AA-AB pairs

If none of the Type A devices in the AA-pair is in the coverage of the Type A device of the AB-pair, these two Type A devices are synchronized to the Type A device of the AB-pair via the Type B device.

If none of the two Type A devices in the AA-pair is in the coverage of the Type B device, these two Type A devices along with the Type A device in the AB-pair synchronize with each other according to Rule no.6.

3.3 Case III: AB-BB Pairs

The logic topology of Case III is shown in Figure 8. Again, the Type B device in the AB pair is forced to synchronize to its Type A master device according to Rule no.7. If any of the Type B devices in the BB-pair is in the coverage of the Type B device in the AB pair, then both of the Type B devices in the BB-pair will synchronize to the Type B device in the AB-pair according to Rule no.3. If the Type A device in the AB pair is also in the coverage of any of the Type B devices in the BB-pair, the Type A device is then forced to synchronize to the Type B devices in the BB-pair according to Rule no.3. If the Type A device in the AB-pair is not in the coverage of any of the Type B devices in the BB-pair, then the Type B devices in the BB-pair indirectly synchronize to the Type A device via the Type B device in the AB-pair.

If none of the Type B devices in the BB-pair is in the coverage of the Type B device in the AB-pair, the Type A device in the AB-pair is forced to synchronize to the Type B devices in the BB-pair according to Rule no.1. The Type B device in the AB-pair then indirectly synchronizes to the Type B devices in the BB-pair via the Type A device.

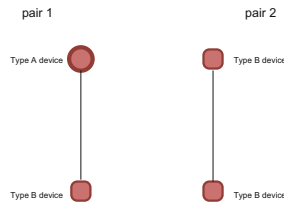


Fig. 8. Case II: AB-BB pairs

3.4 Case IV: AB-AB Pairs

The logic topology of Case IV is shown in Figure 9. In this case, both of the Type B devices are forced to synchronize to their own master Type A devices according to Rule no.7. If none of the Type A devices receives the mode-A0 beacons from each other or from the Type B device in another AB-pair, then the two Type-B devices are the only forced synchronized devices. Based on Rule no.4, these two Type B devices do not synchronize with each other even if they are in the coverage of each other. As a result, the four devices cannot synchronize with

each other, and the interference-free coexistence between these two AB-pairs are compromised.

Note that if any additional link other than the links shown in Figure 9 is established, these four devices will be able to synchronize to each other. In the next section, we will give a thorough mathematical analysis on how frequently the case shown in Figure 9 may occur given that the devices use directional antennas. We will also show how clock drifting affects the synchronization using the OPNET-based simulations.

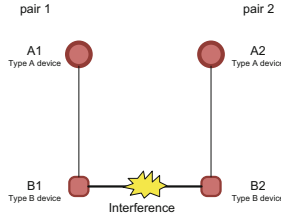


Fig. 9. Case IV: AB-AB pairs

4 Simulation and Analysis

To better evaluate the impact of synchronization on coexistence, we consider realistic network topologies that may lead to the logic topologies illustrated in Section 3. We first consider the scenario shown in Figure 9, as our earlier discussion suggests that there exists a non-zero probability that devices may not synchronize with each other. Our first objective in this section is to evaluate how frequently this scenario can occur by deriving a probability upper bound.

4.1 System Setup

We consider two AB pairs, each forming a MSPr and randomly located in a given area. The transmission range of each Type A device is 10 meters, and the transmission range of each Type B device is 3 meters, based on the ECMA-387 standard. The beam width of the Type A devices varies from 15 to 90 degrees (array antennas), while the beam width of the Type B devices is fixed at 90 degrees (fixed antenna). These settings reflect the differences in device complexity.

The distance between the two Type B devices is randomly distributed between 0 meter to 3 meters so that the Type B devices are in the coverage of each other. The antennas of the two Type B devices are randomly oriented. Each Type A device is located in the coverage of its own slave Type B device. Since we assume that the MSPr has been formed, the antenna of each Type A device is pointed toward the slave Type B devices. The system setup is illustrated in Figure 10.

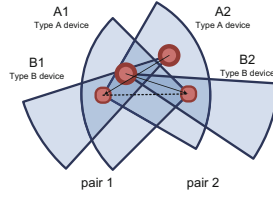


Fig. 10. System setup

4.2 Probability Upper Bound

As mentioned in Section 3, synchronization in Case IV is not a problem as long as there exists a link — other than the link between the Type B devices — between the two AB pairs. Denote the event that there exists a link between device i and j as L_{ij} . The probability that these two AB pairs cannot synchronize to each other can be obtained by

$$P_{no-sync} = P[L_{B_1B_2}] * P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1} | L_{B_1B_2}]. \tag{1}$$

We first calculate the probability $P[L_{B_1B_2}]$. The event $L_{B_1B_2}$ occurs when the two Type B devices point their antennas to each other. Therefore, we have

$$P[L_{B_1B_2}] = P[B_1 \rightarrow B_2] * P[B_2 \rightarrow B_1], \tag{2}$$

where the event $B_i \rightarrow B_j$ represents that Type B device i points its antenna to Type B device j . Given that the antenna of a Type B device is randomly oriented with a beam width of 90 degrees and the location of the Type B devices are randomly selected, one can calculate $P[B_i \rightarrow B_j]$ by

$$P[B_i \rightarrow B_j] = \int_0^{2\pi} \int_{\phi}^{\phi-2\pi} \frac{\pi}{2} \frac{1}{2\pi} d\theta d\phi = \frac{1}{4}. \tag{3}$$

With Eq. (3), the probability $P[L_{B_1B_2}]$ can then be calculated by

$$P[L_{B_1B_2}] = \frac{1}{4} * \frac{1}{4} = \frac{1}{16}. \tag{4}$$

To calculate $P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1} | L_{B_1B_2}]$, we first rewrite the probability as

$$\begin{aligned} & P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1} | L_{B_1B_2}] \\ &= 1 - P[L_{A_1B_2} | L_{B_1B_2}] - P[L_{A_1B_2} | L_{B_1B_2}] \\ &\quad - P[L_{A_2B_1} | L_{B_1B_2}] + P[L_{A_1B_2} \cap L_{A_2B_1} | L_{B_1B_2}] \\ &\quad + P[L_{A_1A_2} \cap L_{A_2B_1} | L_{B_1B_2}] + P[L_{A_1A_2} \cap L_{A_1B_2} | L_{B_1B_2}] \\ &\quad - P[L_{A_1A_2} \cap L_{A_1B_2} \cap L_{A_2B_1} | L_{B_1B_2}]. \end{aligned} \tag{5}$$

By closely examining these events, one can find that only $P[L_{A_1A_2}|L_{B_1B_2}]$, $P[L_{A_1B_2}|L_{BB}]$, $P[L_{A_2B_1}|L_{B_1B_2}]$, and $P[L_{A_1B_2} \cap L_{A_2B_1}|L_{B_1B_2}]$ have non-zero values. In other words, the other events cannot occur with any given orientation and locations of the devices. Thus, Eq. (5) can be further simplified as

$$\begin{aligned} &P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1}|L_{B_1B_2}] \\ &= 1 - P[L_{A_1A_2}|L_{B_1B_2}] - P[L_{A_1B_2} \cup L_{A_2B_1}|L_{B_1B_2}]. \end{aligned} \quad (6)$$

$P[L_{A_1B_2} \cup L_{A_2B_1}|L_{B_1B_2}]$ in Eq. (6) is very difficult to be obtained. Therefore, instead of finding the exact value of $P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1}|L_{B_1B_2}]$, we calculate its upper bound by

$$\begin{aligned} &P[L'_{A_1A_2} \cap L'_{A_1B_2} \cap L'_{A_2B_1}|L_{B_1B_2}] \leq \\ &1 - P[L_{A_1A_2}|L_{B_1B_2}]. \end{aligned} \quad (7)$$

Finally, the upper bound of $P_{no-sync}$ is obtained by

$$P_{no-sync} < \frac{1}{16} * (1 - P[L_{A_1A_2}|L_{B_1B_2}]). \quad (8)$$

Figure 11 illustrates the upper bound of $P_{no-sync}$ for different beam widths of the Type A devices. It can be found that the probability linearly decreases with the increase of the beam width. The result seems contracting the intuition as the larger the beam width, the more chances that there exist additional links between the AB pairs so that devices can synchronize with each other. However, it is noted that the probabilities are plotted based on Eq. (8). Therefore, the larger the beam width, the larger the value of $P[L_{A_1A_2}|L_{B_1B_2}]$, and, thus the smaller the upper bound.

Another important observation in Figure 11 is that the maximum of the upper bound is only 0.06, which occurs when the beam width of the Type A device is 15 degrees. Note that $P_{no-sync}$ is the ‘‘conditional’’ probability that two AB pairs lose the synchronization given that their Type B devices are in the coverage of each other (within a 3 meter range). If we take into account the cases where the Type B devices of the AB pairs are outside the coverage of each other, the probability of two AB pairs losing synchronization with each other (so that they interfere with each other) will be even smaller.

4.3 Simulation Results

We conduct OPNET-based simulations to evaluate the impact of clocking drifting on synchronization in realistic network environments. The synchronization procedure is implemented as follows. When a device receives a beacon from another device, it extracts the timing information in the beacon and determines the start time of the beacon timing structure of the beacon transmitter. By comparing the start time of all neighbors with its own start time, the device can determine the amount of relative drifting and then adjust the local clock according to the rules in Section 2.3. If the device should synchronize to a slower-clock

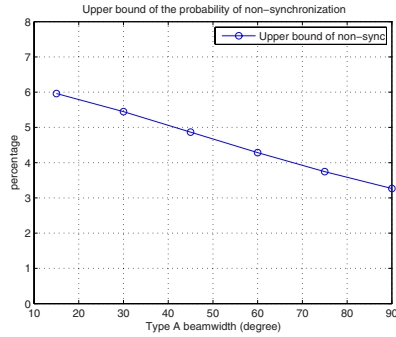


Fig. 11. Probability upper bound: Lack of synchronization in Case IV

device, it delays the start time in his next beacon timing structure. If the device should synchronize to a faster-clock device, it advances the start time in his next beacon timing structure.

We consider Case I and Case II in Section 3. The physical topology and clock drifts for each case are given in the following subsections.

Case I: AA-BB pairs. Figure 12 shows the physical topology and the coverage of each device. Based on the setting, A2 is in the coverage of B1 and B2 (and vice versa). The result network connectivity is shown in Figure 13. The arrows in the figure represent the relation in terms of clock adjustment between devices. An outbound arrow represents that a device may synchronize to the device to which the arrow points. Figure 13 shows that A1 synchronizes to A2, A2 synchronizes to either B1 or B2, and either B1 synchronizes to B2 or B2 synchronizes to B1 depending on their clock drifts. In this simulation, the clock drift of A1, A2, B1, and B2 are randomly set as 9.9 ppm, 4.32 ppm, -4.7 ppm, and 6.35 ppm. The length of the beacon timing structure is set as 256 milliseconds. Therefore, the total clock drift within one beacon timing structure can be calculated as +2.54 microseconds (us) for A1, +1.11 us for A2, -1.20 us for B1, and +1.63 us for B2.

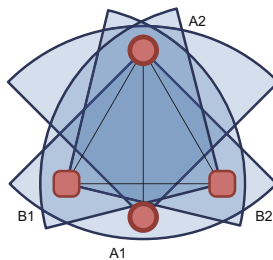


Fig. 12. Topology of Case I: AA-BB pairs

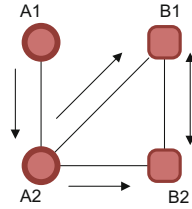


Fig. 13. Logic links and synchronization relationship in Case I

The differences between the start times of the four devices are shown in Figure 14. Here, we use B2's start time as the reference. One can find that the start times of the four devices are not perfectly aligned due to the clock drifting. However, they do not drift away with the time. Figure 14 shows that B1 is faster than B2, but the difference never exceeds 3.2 us. On the other hand, A1 is always slower than B2 but the difference never exceeds 2.2 us. The simulation shows that these four devices are synchronized as expected.

Figure 15 shows the clock adjustment of each device at the beginning of each beacon timing structure. Since the clock resolution is 1 us in our simulation, the adjustment must be a round off of the actual difference of devices's start times. For example, B1 delays its clock either 2 or 3 us in order to synchronize to the slowest device B2. The figure shows that A1 tries to synchronize to A2, A2 tries to synchronize to B1 while B1 tries to synchronize to B2.

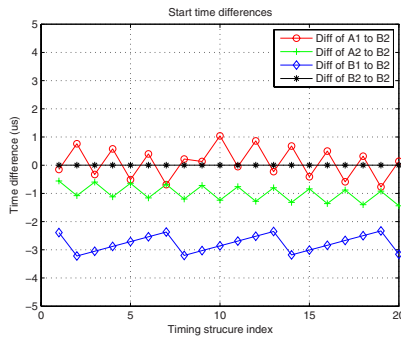


Fig. 14. Start time differences between devices in Case I

Case II: AA-AB pairs. Figure 16 shows the physical topology and the coverage of each device. Based on the setting, A2 is also in the coverage of A3 and B1 (and vice versa). The resulting network connectivity is shown in Figure 17. We can find that in this case, A1 should synchronize to A2, A2 should synchronize to B1 (i.e., the slave of A3), B1 should synchronize to its master device, and finally A3 should synchronize to A2. In other words, every device should adjust

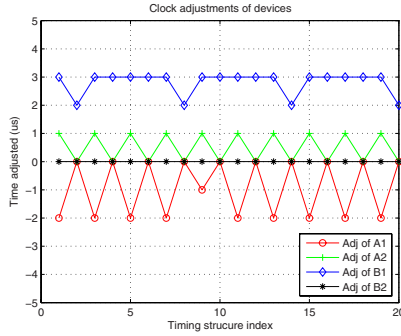


Fig. 15. Clock adjustment in Case I

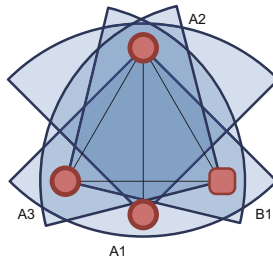


Fig. 16. Topology of Case II: AA-AB pairs

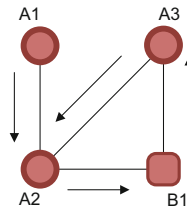


Fig. 17. Logic links and synchronization relationship in Case II

its local clock. This is quite different from the previous case where B2 is the clock reference of all other devices.

In this simulation, the clock drift of A1, A2, B1, and B2 are randomly set as 1.49 ppm, 4.5 ppm, 9.66 ppm, and 7.74 ppm. The length of the beacon timing structure is still 256 milliseconds. Therefore, the total clock drift within one beacon timing structure can be calculated as +0.38 us for A1, +1.15 us for A2, +2.47 us for A3, and +1.98 us for B1.

The differences between the start times of the four devices are plotted in Figure 18. Here, we use A3’s start time as the reference. One can find that the start times of the four devices are again not aligned due to the clock drifting.

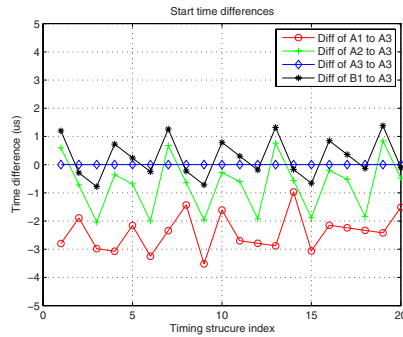


Fig. 18. Start time differences between devices in Case II

However, they still do not drift away with the time even though not a single device (such as B2 in Case II) can be used as the absolute clock reference. We can find that the maximum difference between the devices's start times never exceeds 4 us.

One interesting observation from Figure 18 is that all of the curves change in a similar way. For example, when the difference between B1's and A3's start times decreases/increases (the 'star' line), the difference between A1's and A3's start times (the 'square' line) decreases/increases as well, with a possible delay of 256 ms. The same situation can be found for the difference between A2's and A3's start times. The observation verifies the so-called circular synchronization as we pointed out earlier in this section.

5 Conclusions

In this paper, we investigated the interference among heterogenous devices in the 60 GHz band. The ECMA-387 standard was used as our study case to demonstrate how interference-free coexistence can be achieved without imposing too much complexity on simple devices. The Dual-beacon Protocol of the ECMA-387 protocol was thoroughly investigated, especially from the aspect of synchronization between heterogeneous devices. Our numerical and simulation results show that with a probability of more than 0.94, the heterogenous devices using the ECMA-387 standard can coexist without interfering with each other.

References

1. Lu, M.-H., Steenkiste, P., Chen, T.: Video streaming over 802.11 wlan with content-aware adaptive retry. In: IEEE International Conference on Multimedia and Expo., ICME 2005, July 2005, pp. 723–726 (2005)
2. LG Electronics Inc.: Matsushita Electric Industrial Co., Ltd (Panasonic), NEC Corporation, SAMSUNG ELECTRONICS, CO., LTD, SiBEAM, Inc., Sony Corporation and Toshiba Corporation. WirelessHD Specification Version 1.0 Overview (October 2007)

3. Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs): Amendment 2: Millimeter-wave based Alternative Physical Layer Extension, Draft P802.15.3c/D13 (July 2009)
4. <http://wirelessgigabitalliance.org/>
5. http://www.ieee802.org/11/Reports/tgad_update.htm
6. Eldad Perahia, Intel, TGad Functional Requirements, IEEE 802.11-09/0094r0 (January 2009)
7. Ecma International TC48, High Rate 60 GHz PHY, MAC and HDMI PAL, Standard ECMA-387 1st edn. (December 2008)

AAA-IDEA 2009

Session I – Networking

Optimisation of Power Consumption in Wired Packet Networks

Erol Gelenbe¹ and Simone Silvestri²

¹ Dept. of Electrical & Electronic Engineering, Imperial College, UK
`e.gelenbe@imperial.ac.uk`

² Department of Computer Science, Sapienza University of Rome, Italy
`simone.silvestri@di.uniroma1.it`

Abstract. Over 500 million host computers, three billion PCs and mobile devices consume over a billion kilowatts of electricity. As part of this “system” computer networks consume an increasing amount of energy, and help reduce energy expenditure from other sources through E-Work, E-Commerce and E-Learning. Traditionally, network design seeks to minimise network cost and maximise quality of service (QoS). This paper examines some approaches for dynamically managing wired packet networks to minimise energy consumption while meeting users’ QoS needs, by automatically turning link drivers and/or routers on/off in response to changes in network load.

1 Introduction

Electrical energy is needed for Information and Communication Technologies (ICT) both to operate and cool the equipment. However ICT plays a complex role in energy consumption: as summarised by Francesco Serafini of HP: “ICT covers 2% of the global energy consumption. We can work to halve it to 1%, but the priority is to work to decrease the remaining 98%” (through greater use of ICT), via the “communicate more and travel less” paradigm. (cf. T. Cowen of BT): “if people rely heavily on videoconferencing, distance working, e-learning and e-commerce, there can be a sharp decrease in flights and journeys which will save money and carbon”. Thus computer-communication networks offer a way forward for reducing the consumption of energy and carbon emission by reducing land and air transport, but this potential reduction is partially offset by the power used by data centres and computer networks [1]. Network and web service providers have electrical annual costs in the billions of pounds. Even a fraction of energy savings in networks could lead to reduced financial costs and carbon savings. Therefore *future computer networks should include energy aware traffic management and routing techniques*, together with efficient hardware level energy management, to provide acceptable levels of QoS, at the lowest possible energy levels. The importance of *packet networks* on energy consumption increases with the “convergence to the Internet Protocol (IP)” paradigm whereby most modes of communication, including mobile telephony, are increasingly supported by underlying packet networks. Since IP networks rely on network nodes

and links, the electrical energy used for operating and cooling the equipment creates a crucial need for research on energy saving strategies for networks.

Our research starts from the premise that if one knows (a) the traffic being carried by a network, (b) the Quality of Service (QoS) requirements of the traffic flows, and (c) the capacity and energy requirements of the nodes and links that are available to store and forward this traffic, then (d) one could rationally select a set of nodes and inter-node links, and paths through these selected nodes from source to destination, so as to satisfy the required QoS at a minimum energy utilisation. Although much work has been done for power management in wireless networks, this subject has not yet received much attention for wired networks. In the wired case, the problem we describe can in principle be formulated as a standard network topology optimisation problem where the cost to be minimised is power consumption, and the constraint to be respected is QoS. However a *static optimisation* approach, assuming a given fixed set of traffic flows, is impractical due to dynamic changes in traffic levels. Approaches that can take advantage of short period changes in traffic levels, e.g., hourly or even shorter, would be more useful, and one can also consider variations in electric energy costs. More sophisticated approaches, which we will not study in this project) might consider the impact on carbon emissions. Moving data centres to remote locations where energy is cheap and outside temperatures are low (for cooling) has been considered, but this idea is impractical for networks where much of the infrastructure should be close to the end users.

Our work considers a *dynamic approach* to energy optimisation in packet networks, where link drivers and/or nodes are turned on or off, in response to traffic load in the network, with ensuring changes in the paths followed by the traffic so as to meet the QoS needs of the flows. Since it is technically easier to keep routers “on” constantly, and just turn the links’ drivers on/off according to need, and since drivers can use up to 40% or more of the energy consumed by a network node [2, 3], we will focus in particular on that approach.

Our main objective is therefore to develop a principled approach for the design of a software based “Energy Management System” (EMS) for wired packet networks that will make and carry out dynamic decisions to minimise network energy consumption while respecting QoS constraints. The EMS would run on top of the network (e.g., IP) layer, and *continuously* undertake the following steps:

- a) Observe ongoing traffic flows in the network, monitor the status of nodes and the network’s power consumption.
- b) Select the network configuration that offers an acceptable or better level of QoS to ongoing and predicted flows, with *lower energy consumption*.
- c) Manage and sequence dynamic changes in links and nodes, and reroute traffic, to achieve reduced power consumption at acceptable QoS levels.

This study does not address optimisation related to cooling systems for network routers because it would require knowledge of the physical spaces and external conditions related to buildings where routers are housed. However, when network

equipment consumes less power, there is an induced effect on heat dissipation power used for cooling.

1.1 Prior Work on Power in Wired Networks

Energy saving strategies for wired networks have not received much attention. Some preliminary works can be found in [2, 3, 4, 5, 6, 7]. Paper [4] characterizes the variation due to fluctuating electricity prices and argue that existing distributed systems should be able to exploit this variation for significant economic gains. In [5] some methods that allow for detection of inactivity periods in Ethernet networks are proposed. Such periods are exploited to obtain energy savings with little impact on loss or delay. Papers [2, 6] study the power profile of modern network devices and propose some method in order to reduce the energy consumption without affecting the performance. The authors of [3] considers jobs reallocation in order to reduce the number of active switches into the network. Finally, paper [7] surveys methods and technologies currently used for energy-efficient operation of computer hardware and network infrastructure in the context of cloud computing.

None of the previous works consider a general framework for energy optimisation in packet networks, where link drivers and/or nodes are turned on or off, in response to traffic load in the network, with ensuring changes in the paths followed by the traffic so as to meet the QoS needs of the flows. We introduced this problem in [8].

1.2 Prior Work on Power in Wireless Networks

Differently from the wired case, energy-saving routing protocols in wireless networks have been studied in detail. Energy saving techniques for wireless sensor networks [9], are important because of the limited power availability in networks which operate with batteries or renewable energy sources. “Topology control” (TC) algorithms [10, 11] for wireless networks have been proposed so as to dynamically modify the network graph to maintain or optimize desirable global properties, such as network capacity or user perceived QoS, while reducing energy consumption and wireless interference between nodes. These approaches dynamically adjust the transmission power of each node in order to save energy while guaranteeing connectivity. Another important criterion considered is to limit the ratio of the number of hops traversed by packets from the sources to the destinations, for a given power setting, to the number of hops traversed if all nodes were transmitting at maximum power, and some complex trade-offs occur as the nodes’ transmission power levels are varied. While the maximum power can yield the minimum number of hops, higher power levels will adversely affect collisions and interference, lower levels of transmission power potentially lengthen the paths traversed by packets, but reduce collisions and interference on each hop. As the hop count increases, the energy used per packet can also increase and adverse effects such as delay and loss can also increase.

In [12] topology control is studied with the addition of QoS constraints, and an integer linear programming problem is formulated so as to assign transmission radii to sensors (hence establishing the network topology) so as to minimise the overall global energy consumption while respecting the desired QoS constraints. Although it has similarities to the research we propose, the TC problem for wireless networks cannot be mapped directly into the problem we consider. In a wireless sensor network, a node may change the set of nodes with which it communicates in one hop by adjusting its transmission power, and hence range. This is not the case in wired networks, where a router is connected in one hop to routers which are turned on and connected to it via a link. In a wired node, power consumption may change as a function of the node's workload based on turning some or all the cores in the processors on an off, but this capability may be difficult to control explicitly, and in any case close to 60% of the router's energy consumption typically results from peripheral hardware. In TC for wireless nodes, one cannot switch off some wireless nodes because they may also act as sensors whose role is to gather information, in addition to forwarding packets.

Turning nodes off has been considered in [13] for an energy aware distributed data-centric routing protocol that takes into account the physical location of the sources of data; this is less relevant to the wired networks because wired packet networks may have source and destination nodes anywhere throughout the network. Finally, the broadcast nature of the wireless medium introduces both a degree of freedom (local multicast to neighbours) and additional constraints (interference and collisions) which do not exist in wired packet networks.

In [9] a middleware that manages the state (idle, active, sleeping) of sensor nodes so as to reduce the network's energy consumption, is proposed and two algorithms are considered, either turning off a node when the node is not involved in sending, forwarding or receiving data, or using information about the spatial "density of active nodes" so as to change the proportion of time spent in the active state (duty cycle) of nodes in proportion to the density of active nodes. In a sensor network the volume of traffic is generally relatively low and each node can alternate between different states (idle, active, sleep) quite frequently. This may not be possible in a wired scenario, where traffic volumes and speeds can be high, QoS constraints can be stringent, and large packet loss rates cannot be tolerated. Also, large wired routers can take a non negligible time to be turned on/off, especially when precautions must be taken to avoid packet losses in the queues. In [14] battery powered wireless ad-hoc networks are considered, and a routing technique which selects paths so as to both satisfy QoS constraints and minimises power consumption is suggested, so and some of these ideas may be used in our proposed research.

1.3 Power Minimisation in Clusters of Servers

Recent research has also considered energy savings in clusters of database and web servers, where one seeks to minimise power while guaranteeing acceptable levels of throughput and response time [15]. In such systems, energy consumption depends on CPU utilisation, but is also caused by components such as disks,

memory, network devices, etc., and an idle server may still use up to 60% of its peak power [15, 16]. Thus, in order to minimise energy power, it is necessary to turn off or down complete ICT servers as a function of load. In [16] policies based on economic criteria allocate resources within a large cluster of servers, focusing on energy, by managing the request dispatcher so as to concentrate the incoming load on a minimal set of active servers that fulfill the QoS agreement, while other servers are maintained in a low-power idle state. The related dynamic provisioning problem is studied in [17] for long-lived TPC connections, as in the case of instant messaging (Microsoft Messenger, Skype, etc.) or online gaming, using an approach that consists of a dynamic provisioning algorithm and a load dispatching policy. In [18], a dynamic provisioning technique is proposed for a platform that hosts several applications, using a queuing model for system analysis, and a provisioning technique. The analytical model yields the minimum number of servers required to respect the QoS agreement, and the provisioning technique uses a predictive strategy, based on long time scales, to determine future load. The prediction is then adjusted uses a reactive provisioning policy at small time scales, to address problems that may be caused by sudden changes in system load.

The main difference with the networked context is that the research on data centres addresses the question “What is the least number of processing nodes that are needed?”, while in the packet network context we first identify “Which nodes are needed within the given topology, and then decide how traffic should be re-routed through the active nodes. The networking context is also more complex to manage in the presence of ongoing traffic flows that will need to be re-routed so as to constantly satisfy the QoS constraints.

2 Technical Approach

As indicated earlier, this work focuses on dynamic energy optimization in the context of network routing, with monitoring of the current flows and prediction of future flows in the network, including source and destination pairs, volume of traffic in each flow, QoS constraints per flow, the path for each flow, and different “activity levels” ranging from “sleep” to “fully active” that each of the router nodes can have. Our work therefore addresses:

- (A) The formalisation of dynamic network management for energy optimization, and development of the optimisation algorithms, and
- (B) The design and implementation of the Energy Management System middleware (EMS) and its experimental evaluation in our network test-bed.

2.1 Dynamic Network Management for Energy Optimisation

For (A), we suggested two approaches: (i) the first assumes that the drivers of the links from a node can be turned on or off, resulting in proportional (possibly non-linear) changes in the energy consumption, (ii) the second, less realistic

approach with today’s technology, considers turning on and off certain nodes, with larger changes in energy consumption but at a slower rate than (i), say in the minutes. We think that (i) is more realistic because the node itself remains “awake” but its communication drivers are turned on/off.

Both approaches can be studied using a representation of the network N as a set of bi-directional links and nodes forming a graph. A link (i, j) $i, j \in N$, has a maximum traffic carrying capacity of $C(i, j)$ packets/seconds. At time t , link (i, j) can be in states: $k(i, j, t) \in \{0, 1, \dots, M\}$ where 0 is the “off” state, and M is the state in which the link operates at maximum capacity $C(i, j)$; in practice we can expect that there will be just two such states, 0 and M . In the case (i) a link’s traffic carrying capacity at time t is

$$K(i, j, t) = k(i, j, t)C(i, j)/M$$

For each $k(i, j, t)$ a link will have a power consumption (energy per unit time) of $P(k(i, j, t))$ while a node has a power consumption denoted by $P(i, t)$. Clearly for symmetric links we have $k(i, j, t) = k(j, i, t)$. In the case (ii) the *node* will have just two states, on or off, and we will be dealing at each instant of time with a sub-network $n(t) \subseteq N$, and the network $n(t)$ has an instantaneous power consumption of

$$P(n(t), t) = \sum_{i \in n(t)} P(i, t) + \sum_{i, j \in n(t)} P(k(i, j, t))$$

Let $k(t)$ be the matrix $[k(i, j, t)]_{n \times n}$. At time t , the network carries a set $F(t)$ of flows, each characterized by a source-destination pair, and each flow $f(t) \in F(t)$ has a traffic volume $V(f(t))$ and is of type $T(f(t))$ (e.g., UDP, TCP etc.), with *QoS constraints* $Q(f(t))$ which is typically a vector of numerical values (e.g., delay, loss, jitter) which must not be exceeded if the QoS is to be satisfied. The routing scheme at time t is $R(F(t), n(t), k(t), t)$, which assigns each flow to a sequence of nodes in $n(t)$ (for (i) $n(t) = N$) going from the flow’s source to its destination. Since flows interact with each other via the routing scheme, the *observed QoS* for flow $f(t)$ is $q(f(t), R(F(t), n(t), k(t), t))$, a vector denoting the *observed* or measured QoS for $f(t)$. If the network respects the QoS constraints at time t , then

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)) \quad \forall f(t) \in F(t)$$

meaning that, say, the observed delay, loss and jitter for each flow are smaller than the required upper bound for delay, loss and jitter. The power optimisation problems can then be formulated as follows:

- **For approach (i):** Given a set of flows $F(t)$, find a link-level activity matrix $k(t)$ and an assignment of paths to flows $R(F(t), N, k(t), t)$, such that the QoS constraints are respected:

$$q(f(t), R(F(t), N, k(t), t)) \leq Q(f(t)) \quad \forall f(t) \in F(t)$$

and power consumption $P(N, t)$ is minimised.

- **For approach (ii):** Given a set of flows $F(t)$, find a sub-network $n(t) \subseteq N$ and an assignment of flows to paths in $n(t)$, $R(F(t), n(t), k(t), t)$, such that the QoS constraints are respected:

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)) \quad \forall f(t) \in F(t)$$

and power consumption $P(n(t), t)$ is minimised.

To describe the optimisation problems, we can construct a “network of queues” model [19], where in (i), the link “service rates” are proportional to $k(i, j, t)$, so that algorithms can be designed to choose the matrix $k(t)$ as a function of the flows’ characteristics and QoS constraints so as to minimise power consumption. A differentiable cost function will be developed to seek local minima with gradient-type techniques to seek local minima, with the intention of deriving expect fast (polynomial time in the size of the network) optimisation algorithms using our experience with gradient descent learning in neural networks [20]. For (ii), to avoid the combinatorial explosion in seeking the best choice among all subsets of active nodes, we will examine greedy heuristics to identify the links and nodes that carry the least traffic, or the links and nodes carrying the least traffic among those with the highest power consumption, for possibly being turned off, and examine incremental policies for turning links or node off or on. This raises a challenging “real-time” aspect further discussed below.

As we change the network topology from network $n_1(t)$ to another one $n_2(t')$, $t' > t$, these networks will necessarily have common nodes (to avoid losing traffic), and hence traffic re-routing must accompany changes in network topology. Thus re-routing must be carried out in stages, so that once the future sub-network $n_2(t')$ is selected then:

- First certain flows should be diverted to nodes which are common to the current network and the future network, to avoid traffic loss during re-routing,
- Then certain links or nodes are turned off, while others may be turned on, so that the new network is established.
- Finally traffic has to be re-routed to take its new form.

These steps must be taken in a manner that the QoS for each flow is respected and the overall integrated power consumption over all the steps is lower. In networks operated by Internet service providers, excess capacity is usually maintained so as to protect the users’ traffic in the case of node and link failures of nodes and this feature actually simplifies the practical implementation of our proposal. We will therefore examine how this feature can be used to advantage in both (i) and (ii), as traffic is re-routed so as to turn off certain links or nodes to reduce power consumption.

2.2 Design of the EMS

The EMS, the software system which will implement the optimisation algorithms, should use a proactive approach to monitor the network and explore

control decisions. It will collect information about system state, then compute a course of action, and decide on the sequence of changes to be made. The EMS will gather information about: 1. *Active traffic flows*: sources and destinations, information about paths, traffic rates, experienced QoS (delay, jitter, packet loss), 2. *Power consumption*: instantaneous power consumption of each link driver and router, 3. *Router traffic*: identity (flow) and amount of traffic passing through each router.

The EMS uses the optimisation algorithms that we develop to evaluate whether a change in the network can provide significant reduction in power consumption with acceptable QoS, and evaluate whether the sequence of actions to reroute traffic and turn off and on certain (i) links (drivers) and/or (ii) routers can be carried out without major QoS degradation. Then it acts upon the routing tables to redirect traffic, then turn some drivers and nodes off, and it might turn others on, then again re-redirect traffic, while continuing to monitor the QoS and power consumption.

The role of the EMS bears some similarity with previous work on the Cognitive Packet Network (CPN) routing protocol [21] that monitors the QoS state of flows, and recommends the choice of paths to optimise users' QoS.

CPN is implemented in the Imperial College test-bed described in <http://san.ee.ic.ac.uk>, using “smart packets” (SP) which travel through the network from node to node to collect QoS information at nodes and links. The resulting information is then returned to the sources, using “acknowledgement packets” (ACKs) which differ from conventional ACKs in networks. In CPN, the traffic payload is carried by “dumb packets” which use source routing. In CPN decisions are distributed, and sources select the paths that they use. In EMS on the other hand, we will begin with a *centralised* approach, even though a distributed approach may be taken at a second stage. We will also consider how the EMS may combine “admission control” for new connections to power optimisation and traffic routing. However, once the EMS determines the links or nodes to be turned off, and chooses the new paths for the flows that are affected by this change, it can inform the source nodes of the flows and provide them directly with the new source routes before it actually makes the changes at the link or node level. Like CPN, EMS may also use SPs to (I) collect QoS information about nodes and paths, and (II) collect data about ongoing energy consumption at nodes. The latter may be done with sensors placed at the power plugs of drivers or nodes or from their power supplies. Using the same SP to collect both QoS data and power data would also provide better timing accuracy about the network state which is of interest and includes both power and QoS. Once the best options have been chosen and the sequence of decisions are selected, the EMS can also distribute the routing decisions throughout the network using a mechanism similar to MPLS, or use source routing as indicated earlier.

3 Some Preliminary Experiments

In order to illustrate the feasibility of our research, we ran some simple experiments on the test-bed of Figure 1 representing the SwitchLAN network topology

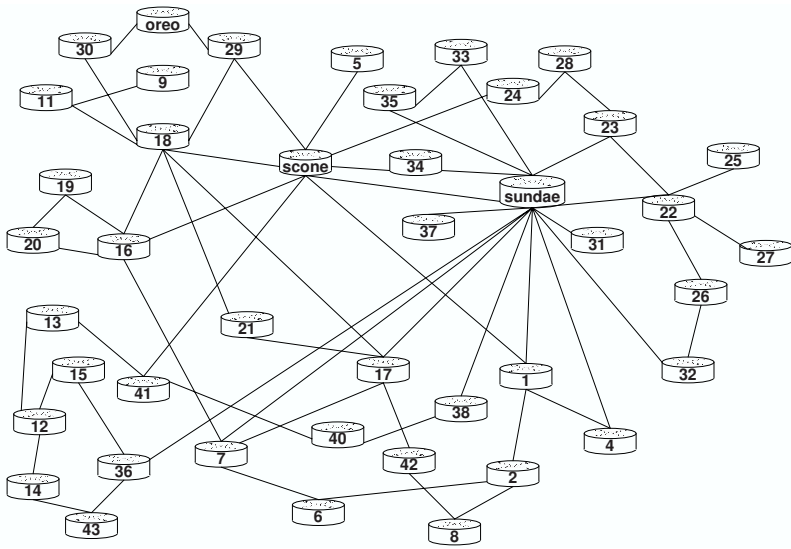


Fig. 1. Topology of the test-bed in use

[22]. The network is composed by 44 routers running the CPN adaptive routing protocol [21]. We emulated the effect of disabling the drivers of some of the nodes' links by blocking their links.

The first experiment aims at studying QoS metrics of interest, namely delay, packet loss and jitter, while turning some routers off. We activated three 1Mb/s flows from node 8 to 1, 13 to 40, 30 to 21. Every 100 seconds we deactivated one router by disabling its links. In particular we deactivated nodes 28, 2, 34, 41, 16, 18, 26, emulating a 10-20% reduction in power consumption. Figures 2, 3 and 4 show the measured delay, packet loss and jitter, respectively.

These results highlight that not all topology changes affect the perceived QoS. In particular, changes made at instants 100s, 300s, 500s and 700s do not have a particular influence on the delay and packet loss, since the paths used by the routing protocol do not contain any of these nodes. By contrast, changes made at instants 200s, 400s, and 600s result in an increase of the measured delay and in momentary peaks of packet loss. Delay increases because these nodes lie on the shortest path among sources and destinations, thus longer paths have to be used at each change. The reason of the peaks in packet loss is instead twofold. On the one hand, some packets are still stored in the router queues at the moment in which links are disabled. On the other hand, the CPN protocol needs to detect the change and eventually discover alternative paths to the destination. Figure 4 shows that jitter remains unchanged in this experiment.

From these results it is possible to deduce that our research can effectively reduce the energy consumption, while guaranteeing an acceptable level of QoS, by turning on and off routers according to the network load.

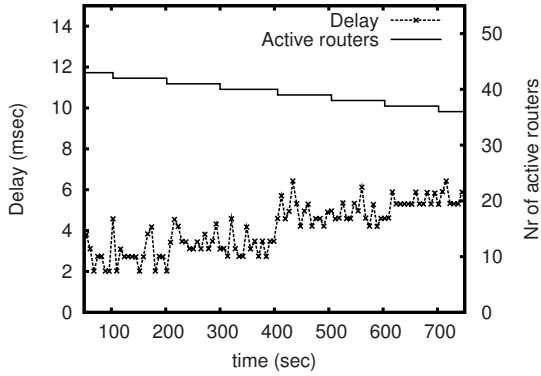


Fig. 2. Delay - first experiment

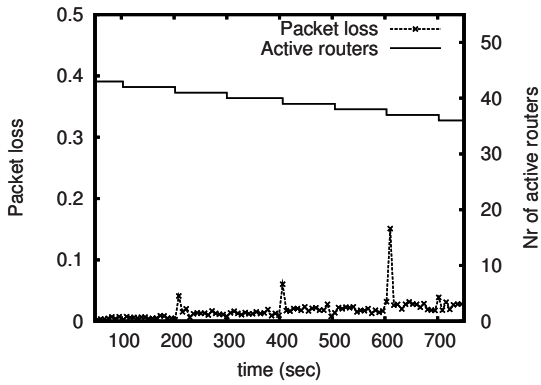


Fig. 3. Packet loss - first experiment

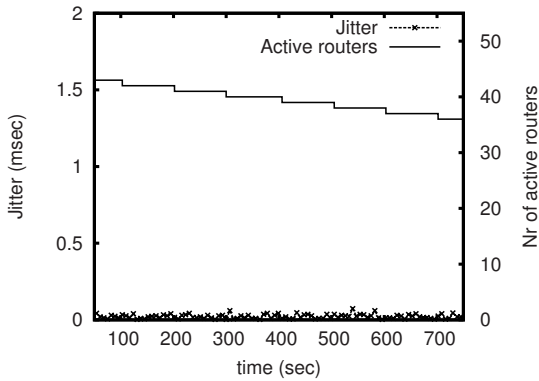


Fig. 4. Jitter - first experiment

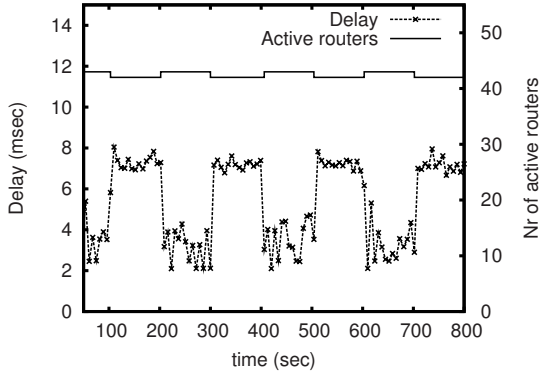


Fig. 5. Delay - second experiment

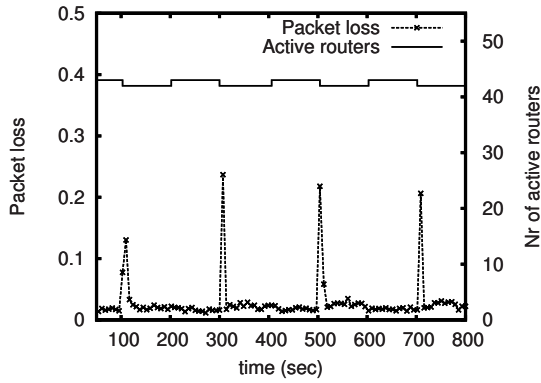


Fig. 6. Packet loss - second experiment

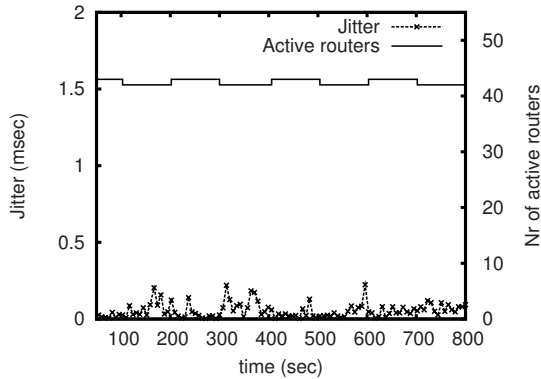


Fig. 7. Jitter - second experiment

The second experiment aims at showing the CPN protocol capabilities of reacting in consequence of a change in the network. We activate three 1Mb/seconds flows from node 23 to 38, 17 to 22, 32 to 33, and we turn *sundae* off (by disabling all of its links) and then on, every 100 seconds. Each flow has a shortest path through *sundae*, thus every time it is turned off the CPN protocol has to adapt the routing scheme to the new topology by using longer paths. Figures 5 and 6 show the measured delay and packet loss, respectively. The network runs CPN which adapts the paths so as to avoid *sundae* when it is off. Although delay and loss increase, they recover rapidly after each change, while jitter appears to remain stable.

4 Conclusions

This paper describes some preliminary results on research that can contribute to discovering principled techniques that can reduce energy consumption in packet networks. It is therefore hoped that this work can influence the ICT research community and industry to direct more attention to these issues.

As a consequence, energy optimisation can become a new and important area of network optimisation and management that directs researchers' attention to energy savings in computer networks.

References

1. Fan, X., Weber, W.-D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: ISCA 2007: Proceedings of the 34th annual international symposium on Computer architecture, pp. 13–23. ACM, New York (2007)
2. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: Power awareness in network design and routing. In: Proceedings of the IEEE International Conference on Computer and Communication, Infocom (2008)
3. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: Energy aware network operations. In: Proceedings of the IEEE Global Internet Symposium (2009)
4. Qureshi, A., Weber, R., Balakrishnan, H., Gutttag, J., Maggs, B.: Cutting the electric bill for internet-scale systems. In: Proceedings of ACM Sigcomm (2009)
5. Gupta, M., Singh, S.: Using low-power modes for energy conservation in ethernet lans. In: Proceedings of the IEEE Infocom Minisymposium (2007)
6. Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., Wright, S.: Power awareness in network design and routing. In: Proceedings of the IEEE International Conference on Computer and Communication, Infocom (2008)
7. Berl, A., Gelenbe, E., di Girolamo, M., Giuliani, G., de Meer, H., Dang, M.Q., Pentikousis, K.: Energy-efficient cloud computing. *The Computer Journal* (August 2009)
8. Gelenbe, E., Silvestri, S.: Reducing power consumption in wired networks. In: Proc. 24th Annual Int. Symp. Computer and Information Sciences, ISCIS 2009 (2009)
9. Xu, Y., Heidemann, J., Estrin, D.: Adaptive energy-conserving routing for multi-hop ad hoc networks. USC/Information Sciences Institute, Research Report 527 (October 2000), <http://www.isi.edu/~johnh/PAPERS/Xu00a.html>

10. Rajaraman, R.: Topology control and routing in ad hoc networks: a survey. *SIGACT News* 33(2), 60–73 (2002)
11. Santi, P.: Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.* 37(2), 164–194 (2005)
12. Jia, X., Li, D., Du, D.: Qos topology control in ad hoc wireless networks. In: *Proc. of IEEE INFOCOM 2004* (2004)
13. Boukerche, A., Cheng, X., Linus, J.: A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks, vol. 11(5), pp. 619–635. Kluwer Academic Publishers, Hingham (2005)
14. Gelenbe, E., Lent, R.: Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks* 2(3) (2004)
15. Economou, D., Rivoire, S., Kozyrakis, C., Ranganathan, P.: Hardware-agnostic full-system power modeling. In: *MOBS* (2006)
16. Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P.: Managing energy and server resources in hosting centers. In: *SOSP 2001: Proceedings of the eighteenth ACM symposium on Operating systems principles*, pp. 103–116. ACM, New York (2001)
17. Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., Zhao, F.: Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: *NSDI 2008: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 337–350. USENIX Association, Berkeley (2008)
18. Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., Wood, T.: Agile dynamic provisioning of multi-tier internet applications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 3(1) (March 2008)
19. Gelenbe, E., Pujolle, G., Nelson, J.C.C.: *Introduction to queueing networks*. John Wiley Ltd., New York (2000)
20. Gelenbe, E., Hussain, K.: Learning in the multiple class random neural network. *IEEE Trans. on Neural Networks* 13(6), 1257–1267 (2002)
21. Gelenbe, E., Lent, R., Nunez, A.: Self-aware networks and QoS. *Proceedings of the IEEE* 92(9), 1478–1489 (2004)
22. The Swiss Education & Research Network, <http://www.switch.ch/network/>

Revisiting a QoE Assessment Architecture Six Years Later: Lessons Learned and Remaining Challenges

Amy Csizmar Dalal*

Carleton College, One North College St., Northfield MN 55057, USA
adalal@carleton.edu

Abstract. In 2003, we presented an architecture for a streaming video quality assessment system [1]. Six years later, many of the challenges outlined in that paper remain. This paper revisits the 2003 architecture, updates it given what we have learned in our experience thus far with developing the architecture, and discusses in detail the remaining challenges to the realization of this architecture. We conclude with suggestions for moving beyond the biggest challenges, namely cooperation among the interested parties and system scale.

Keywords: Quality of Experience (QoE), Quality of Service (QoS), Streaming Media, Measurement, Performance, Reliability.

1 Introduction

In 2003 we published a paper [1] proposing a new architecture for assessing users' quality of experience (QoE) of streaming video, video sent on-demand over unicast from a media server to one or more media clients. At the time, RealPlayer was the dominant mechanism for video delivery over the Internet, and the amount of video traffic was a small but increasing fraction of overall Internet traffic [2]. Today, video makes up about 30% of Internet traffic [3], with much of the video embedded on sites such as YouTube and Hulu [4]. Then and now, content providers, content distributors, and ISPs are interested in determining how current network conditions, such as packet losses and delays and available bandwidth, affect the performance, as perceived by the end-users, of streaming video applications, and vice versa. It is this user perception, after all, that drives Internet, application, and content use.

The notion of QoE is a nebulous one: there is no one accepted definition for QoE of streaming video, nor an accepted standard of streaming video QoE measurement. While subjective approaches, such as the Mean Opinion Score [5], are the most natural choice, they suffer from scalability and context accuracy issues. Indeed, a survey of the literature shows that QoE has been measured using

* This work is sponsored by grants from the Howard Hughes Medical Foundation and from Carleton College. Early versions of this work were sponsored by Hewlett-Packard Laboratories.

network-level measurements [6,7,8], frame rate [9], packet-level statistics [10], received signal strength indicators [11], and a complex combination of network, application, and content measurements [12].

The breadth of QoE measurement approaches speaks to the complexity of the QoE measurement problem. Yet we must overcome these challenges to create robust, dependable, efficient, and effective QoE measurement systems. Such systems will move the current trial-and-error approach of network provisioning, application deployment, and protocol design and development to a place where application performance combined with knowledge of network conditions leads to a more responsive approach to video delivery over the Internet, and to future Internet support of rich media applications, such as high-definition video for entertainment, distance learning, telemedicine, and telepresence.

In this paper, we revisit our proposed architecture for a QoE assessment system for streaming video. The main focus of our work to date has been on developing the measurement tool and appropriate QoE measurement(s). We discuss the challenges in developing a QoE measurement, revisit the original architectural goals, update the architectural design, and discuss the remaining challenges. Many of the system-wide challenges that we wrote about in 2003 still exist, and some of them, like garnering cooperation among the interested parties, provide significant barriers to realizing this architecture.

We start in Section 2 by discussing the original architecture goals and describing the motivations and barriers to cooperation of the involved entities. Section 3 highlights the process and remaining challenges in selecting a QoE measurement. Section 4 updates the architecture and discusses the new design elements, including the design of a new key aspect of our system, the health monitor. Section 5 concludes by discussing the key remaining challenges and proposing mechanisms for moving towards the realization of this architecture.

2 Original Architecture

Figure 1 illustrates our original QoE assessment architecture, which focuses on four main goals:

1. **Flexibility:** Measurements should be collected automatically and with a minimal amount of end-user participation. Additionally, the architecture should support on-demand, non-disruptive measurements from any subset of participating hosts, either for diagnostic or planning purposes. The assessment servers, which control the collection of data from media clients and helper agents and coordinate the analysis of this data, provide this functionality within the architecture.
2. **Support for multiple consumers of assessment:** Multiple parties have a vested interest in supporting and delivering streaming video. A stream quality measurement system should provide a mechanism for these parties to fully participate in the system by both sharing and receiving data from other parties. The report servers, which disseminate analyzed data, accomplish this architectural goal.

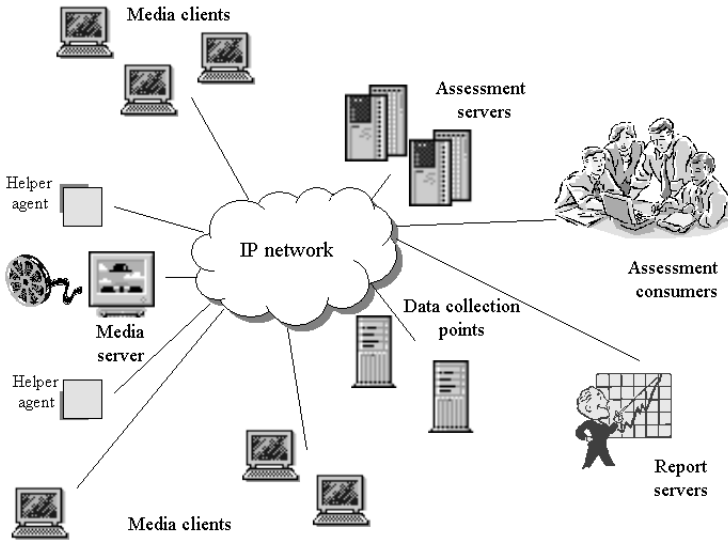


Fig. 1. The original QoE assessment architecture, from [1]

3. **Utilization of existing infrastructure:** A large-scale QoE assessment architecture should leverage existing applications and infrastructure wherever possible, including other measurement infrastructures and existing tools, to reduce the burden on end-users and the network. The proposed architecture accomplishes this by the use of helper agents, which may be existing measurement points within the network, and by collecting measurements from the media player applications (see Section 3).
4. **Responsiveness:** QoE measurements should be used to modify how the system behaves. Examples include determining whether network or server resources should be reprovisioned, whether a stream should be served from a different location on the network, or whether a server should temporarily decrease or increase its sending rate. In the proposed architecture, the report servers and assessment servers collaborate to make this happen.

A QoE assessment system for streaming video necessarily involves independently-operating independently with common needs and goals. **Media servers**, for instance, are most interested in knowing if they are streaming at a bandwidth that their clients can support. **Media clients**, on the other hand, want assurance that their stream quality will either not degrade, if it is currently acceptable, or improve soon, if it's currently less than acceptable. They may also want to know if another media server can deliver the same content at a better quality level. **Content distributors** are interested in content placement for optimal performance, as well as how many servers are needed and whether transcoding of content is necessary. **Network operators** want to know if their networks are healthy, which paths are over- or under-provisioned, whether better routes exist to certain destinations,

and whether reprovisioning of resources is necessary. **Network architects** are interested in how to provision and design networks to best support video traffic, including router placement and peering relationships. Finally, **protocol developers** and **application designers** are interested in how to best utilize existing infrastructure to maximize performance, and in knowing what network and/or application conditions are most likely to affect performance.

There are points along the delivery system where cooperation is required. A content distributor cooperates with network operators and ISPs in order to determine where to locate content servers. Media servers and media clients interact with ISPs to connect to the Internet. Other cooperative relationships, if they existed, would prove beneficial to both parties. Application designers could work with network operators to improve how applications utilize existing network infrastructure. Similarly, network architects could utilize feedback from media servers and clients to improve the resource distribution of future networks.

Historically, cooperation among these parties has been limited. Given a set of data from, for example, a network provider, it is often trivial to determine the source, destination, and/or nature of the data traffic. This opens up both privacy and trade secret issues: customers may not want their competitors to know what sites they are visiting; an outsider may be able to infer the topology of a network, or an ISP's peering agreements; knowledge of an ISP's customers may violate privacy agreements and give its business rivals a competitive advantage. Measurement data might also be used to design a more effective denial of service attack against an ISP or set of media servers. Finally, there is the fear that measurement data might demonstrate that an ISP's service level agreement is not being met for one or more of its customers, opening it to lawsuits for breach of contract. These represent real barriers to cooperation, and we recognize that these issues are non-trivial and difficult to fairly address. We return to this point in Section 5 and present several ideas for mitigating these barriers.

3 QoE Measurement

A key challenge in the development of a QoE measurement architecture is defining QoE for streaming video. A survey of the literature yields several definitions: video distortion as seen by the end user [13]; the number and severity of impairments caused by transmission parameters such as noise [14]; how a user would rate the subjective quality of the stream as compared to other streams s/he has viewed in the past [15,6]; how close the video quality is to satellite or cable video quality [8]. To date, there is no measurement standard for QoE for streaming video, as we discuss in Section 1.

Our approach is to exploit the ease of collection and analysis of objective measurements and infer the user's experience by collecting measurements as close to the user as possible, at the application layer, using an instrumented version of a media player application [15]. The application layer measurements are composed of stream state information reported by the player, which we poll once per second, on quantities such as current average bandwidth, number of lost and retransmitted packets, frame rate, and number of times buffered. They reflect the current

state of the video stream as well as the media server’s response to network conditions, such as packet losses. From this state information, we can infer what a user “sees”, particularly if we know the expected state of a “normal” stream.

As we discuss in [10], not all state measurements are created equal: some of them, such as retransmitted packets, are early indicators of the presence of network congestion; while others, such as average bandwidth, are lagging indicators of network congestion.¹ Because a goal of our architecture is to *predict* streaming video QoE, we focus on collecting and analyzing leading indicators, specifically the number of successfully retransmitted packets.

Discerning the QoE of a video stream requires knowledge of the QoE of past streams. Users are likely to assign similar QoE ratings to streams with similar stream state characteristics. Thus, a measurement system could compare past stream state data and QoE ratings against stream state measurements of a currently-streaming video to assign a QoE rating to that video. Data mining techniques can be used to efficiently search this historical data to find the closest matching stream, and assign a QoE rating to the current stream based on the rating of the closest match. This is the idea behind our own QoE measurement strategy. As a proof of concept, we collected MOS-like measurements and stream state measurements for 228 streams and applied a nearest-neighbor data mining technique to this data to predict stream QoE ratings. Our experiments showed that this approach results in correct QoE ratings assignments between 70 and 90% of the time; see [15, 16] for more details.

Collecting measurement data on a large enough scale to demonstrate the accuracy of various proposed QoE measurements remains a challenging task; most existing data results from smaller experiments on testbed networks, with possibly unrealistic network conditions. There are also questions as to whether a generic QoE measurement can exist for different video scenarios, such as live versus on-demand content or streaming versus progressive download. If different QoE measurements are required by different applications, the architecture must be nimble enough to switch between these measurements and analyses on the fly. Finally, there is the question as to whether measurement and analysis can be completed on sufficient time scales to allow affected parties to react and respond to the results. Our results in [16] show promise, but further work must be done for this architecture to prove viable.

4 System Architecture

Figure 2 updates the architecture discussed in Section 1 based on our experience in developing parts of the system architecture. The new architecture better reflects the complex relationships along the critical path of measurement, and better meets the goals of responsiveness and support for multiple consumers of

¹ To some extent, whether a measurement is a leading or lagging indicator of network congestion depends on the state information that the media player reports. Windows Media Player, for instance, reports the number of application packets that were successfully retransmitted, while QuickTime does not.

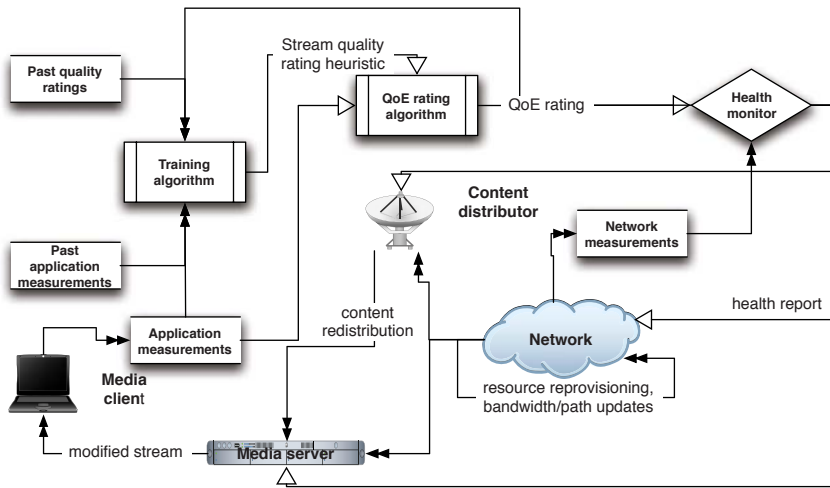


Fig. 2. The revised streaming video QoE assessment architecture. Forward paths are denoted by lines with white arrows, while feedback paths are denoted by lines with double black arrows.

assessment. This version of the architecture also focuses more specifically on the QoE measurement mechanism.

The revised architecture introduces the QoE rater and the health monitor, which utilize current and past measurements to infer current QoE and system health levels, respectively. The QoE rating algorithm relies on past and current application measurements, as well as past QoE ratings, to assign QoE ratings to current streams. The training algorithm uses past QoE and stream state measurements to develop a heuristic by which to evaluate the QoE of current streams, which is used by the QoE rater. The health monitor analyzes the QoE ratings, as well as current network measurements, and periodically sends out status reports to the system entities. The health monitor can also send more frequent updates in cases where it detects that the health of the system is deteriorating from an acceptable level.

Each entity acts upon the information received from the health monitor however it sees fit. Each entity can also request information from other parties in the system. For instance, content distributors may match up health monitor reports with available bandwidth reports from network operators to determine when and how to redistribute content. Media servers might also utilize bandwidth and path updates from network operators to optimize their streaming rates. As updated application and network measurements are fed back into the system, the system becomes self-supporting, reflecting both the current state of the network and the relevant history of the network and application states, similar to the system proposed in [17].

The RTP feedback architecture [18], RTCP, shares many of the same goals we seek here, and yet has historically suffered from scalability and usability

issues. Unlike RTCP's feedback mechanism, which originates solely from the client, feedback in our architecture originates from multiple sources, allowing for a more holistic consideration of system state. Also, while RTCP sends all reports upstream, our architecture allows for multiple feedback and feed-forward paths, to allow for continuous monitoring and updating of system state.

Because the health monitor and QoE rater constantly respond to new measurement data, the system necessarily evolves as application and network data evolve, and as traffic patterns and application usage patterns change. Thus, the system can support both current and future streaming video applications and traffic patterns, and can evolve along with the Internet and its applications.

While the structure of the health monitor is quite simple, its implementation poses several key challenges. Defining an appropriate time-scale for measurement reporting requires a tradeoff between accuracy and relevancy: the monitor needs to distinguish between normal and troublesome network behavior and recognize diurnal and weekly patterns in the data, but must also be nimble enough to respond to sudden changes in network state, such as router outages or server overload, and sudden changes in QoE levels, such as when a media server goes down. We propose a two-pronged strategy. The health monitor receives periodic, time-averaged measurements from the network under "normal" circumstances: for example, packet loss rates over the last five minutes. This reduces unnecessary communication between the network measurement entity and the health monitor. Instantaneous, more frequent measurements are allowed under certain conditions, determined either by the network measurement entity, the health monitor (in the case where QoE ratings drop sharply, for instance), or both. Identifying adequate time periods for more frequent measurement could be determined on a case-by-case basis or by trial and error among the parties. This strategy prevents the system from being overly sensitive and from being unresponsive. An additional challenge is involved in determining when to examine network measurement data and/or QoE ratings more closely. A simple solution is to only consider network measurements when the QoE rating itself is low, borderline, or decreasing from a previous steady state; or, conversely, to consider QoE levels only when network measurements imply declining network conditions. If the QoE measurement, or the network state, is acceptable and stable, no further action needs to be taken.

5 Remaining Challenges: Cooperation and Scale

In previous sections, we have discussed the evolution of our streaming video QoE assessment architecture in light of our experiences developing and realizing the architecture. So far, our work has successfully addressed two of the largest challenges: identifying an appropriate QoE metric and developing a means for analyzing QoE-related data. In order to fully realize the architecture, two key challenges remain: garnering cooperation among independent entities, and determining an appropriate system scale. We address these remaining challenges in this section.

Garnering cooperation entails addressing the main concerns about sharing data addressed in Section II: privacy, security, trade secrets, and legal issues. Solutions already exist to deal with anonymizing and sanitizing network measurement data, although these may be imperfect [19]. Safeguards such as lifetimes could be imposed on shared data, after which the data could only exist in summary form, if at all. Standards of conduct, such as the best practices proposed in [20], should be more widely accepted and implemented. Legal agreements can be brokered between the key parties to address the legal and trade secret issues that can occur in the act of sharing data. For instance, ISPs, in exchange for sharing available bandwidth information with content providers, will receive data from the content providers and from the system to allow it to reprovision itself more efficiently. Content distributors, in exchange for sharing content and encoding information with various parties, would in turn receive path availability information from the networks that would allow them to better provision and place their servers.

Scale is another key challenge in the realization of this architecture. While ideally such an architecture would exist on a nationwide or international scale, in reality it is easier to garner cooperation and marshal the necessary resources on a smaller scale—for example, a content distribution network and its partners. Such a system could leverage the existing business relationships between content providers, the media servers that deliver the content, the connected ISPs and their network operations, and the content distribution networks, allowing for a more natural development of trust. Such closed systems, particularly early on in the deployment of this architecture, could very well provide the needed proof-of-concept for the viability and value of this architecture, paving the way for more wide-scale implementation, perhaps regionally or nationally.

As our experience over the past six years, and the work of the networking community for over a decade, has demonstrated, developing and implementing QoE assessment mechanisms for streaming video remains a difficult and sometimes vexing problem. We have made some key strides, particularly in identifying application-level measurements from which QoE can be inferred, but still need to find viable solutions for the cooperation and scale issues. While the challenges remain great, the payoff of a self-supporting, responsive, and self-healing system for streaming video delivery, in which the complex interactions between network and application performance are better understood, and in which this knowledge is leveraged to design and implement better networks, applications, and protocols, will be greater still.

References

1. Csizmar Dalal, A., Perry, E.: A new architecture for measuring and assessing streaming media quality. In: Proceedings of PAM 2003, La Jolla, CA (April 2003)
2. Li, M., Claypool, M., Kinicki, R., Nichols, J.: Characteristics of streaming media stored on the Web. *ACM Transactions on Internet Technology (TOIT)* 5(4), 601–626 (2005)

3. Business Wire: Ellacoya data shows web traffic overtakes peer-to-peer (p2p) as largest percentage of bandwidth on the network (June 18, 2007)
http://www.businesswire.com/portal/site/google/index.jsp?ndmViewId=news_view&newsId=20070618005912&newsLang=en
4. comScore, Inc.: YouTube attracts 100 million U.S. online video viewers in October 2008 (December 9, 2008),
<http://www.comscore.com/press/release.asp?press=2616>
5. P.910, I.T.R.: Subjective video quality assessment methods for multimedia applications. Recommendations of the ITU, Telecommunications Sector
6. Calyam, P., Sridharan, M., Mandrawa, W., Schopis, P.: Performance measurement and analysis of H.323 traffic. In: Proceedings of the 2004 Passive and Active Measurement Workshop, Antibes Juan-les-Pins, France (April 2004)
7. Cole, R.G., Rosenbluth, J.H.: Voice over ip performance monitoring. SIGCOMM Comput. Commun. Rev. 31(2), 9–24 (2001)
8. Tao, S., Apostolopoulos, J., Guerin, R.: Real-time monitoring of video quality in IP networks. IEEE/ACM Transactions on Networking 16(5), 1052–1065 (2008)
9. Wang, Y., Claypool, M.: RealTracer - tools for measuring the performance of RealVideo on the internet. Kluwer Multimedia Tools and Applications 27(3) (December 2005)
10. Csizmar Dalal, A., Kawaler, E., Tucker, S.: Towards real-time stream quality prediction: Predicting video stream quality from partial stream information. In: Proceedings of The Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), Las Palmas de Gran Canaria, Spain (November 2009) (to appear)
11. Li, M., Li, F., Claypool, M., Kinicki, R.: Weather forecasting - predicting performance for streaming video over wireless LANs. In: Proceedings of NOSSDAV, Stevenson, Washington (June 2005)
12. Gulliver, S.R., Ghinea, G.: Defining user perception of distributed multimedia quality. ACM Trans. Multimedia Comput. Commun. Appl. 2(4), 241–257 (2006)
13. Babich, F., D’Orlando, M., Vatta, F.: Video quality estimation in wireless ip networks: Algorithms and applications. ACM Transactions on Multimedia Computing 4(1) (January 2008)
14. Boutremans, C., Iannaccone, G., Diot, C.: Impact of link failures on VoIP performance. In: NOSSDAV 2002: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, pp. 63–71. ACM, New York (2002)
15. Csizmar Dalal, A., Musicant, D.R., Olson, J., McMenamy, B., Benzaid, S., Kazez, B., Bolan, E.: Predicting user-perceived quality ratings from streaming media data. In: Proceedings of ICC 2007, Glasgow, Scotland (June 2007)
16. Csizmar Dalal, A.: User-perceived quality assessment of streaming media using reduced feature sets. Technical report, Carleton College (April 2009)
17. Allman, M., Paxson, V.: A reactive measurement framework. In: Proceedings of the Passive and Active Measurement Conference, Cleveland, OH (April 2008)
18. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A transport protocol for real-time applications. IETF RFC 3550 (July 2003)
19. Pang, R., Allman, M., Paxson, V., Lee, J.: The devil and packet trace anonymization. Computer Communication Review 36(1) (January 2006)
20. Allman, M., Paxson, V.: Issues and etiquette concerning use of shared measurement data. In: Proceedings of the ACM Internet Measurement Workshop, San Diego, CA (October 2007)

Efficient Authenticated Wireless Roaming via Tunnels

Andreas Noack

Horst Görtz Institut für IT-Sicherheit
Ruhr University Bochum

Abstract. Wireless roaming means that a mobile device is able to switch from one network cell to another while keeping the link to active services. Recent researches [13] showed that it increases the security to establish an authenticated and confidential tunnel directly to a home network which then acts as service provider respectively proxy server for further external services. In this paper we extend the trust assumptions and formal security goals for *wireless roaming via tunnels* (WRT) that were given by Manulis et al. [7].

Additionally, we propose an efficient protocol that realizes the authentication and key agreement for establishing the secure tunnel, whereby considering the delay restrictions that are given by current multimedia services like VoIP or video streaming.

Furthermore we discuss the accounting problem and present a solution that ensures a fair accounting for the foreign network.

Keywords: Wireless networks, security, key agreement, mutual authentication, accounting.

1 Introduction

Wireless LAN is a very popular communication medium today, since it allows its users to be mobile while having access to all services they usually use in a wired LAN. Recent technologies like IEEE 802.11a/g/n also allow a very high bandwidth, so that the advantages from the wired alternative become smaller and smaller.

To let wireless LAN become even more attractive, the coverage has to be improved further on, so that everyone has everywhere access to his preferred services. Of course, it is not possible to realize a single wireless LAN that covers a whole city region. That means, it is necessary to work with several smaller wireless networks that may be operated by foreign network providers. Therefore a cooperation with foreign network providers is required.

There are three problems to solve:

1. When connecting to foreign wireless LAN providers, it is important to preserve the own security.
2. While switching between two wireless LAN cells, current running services like VoIP, video or audio streaming should not be affected.

3. The foreign wireless LAN provider clearly wants to get paid for the service he provides; that means, a fair accounting must be arranged.

Imagine a whole city covered with wireless nodes from private users. Most of them have a direct connection to the internet and are able to distribute their internet link over wireless LAN. There are several companies which want to provide seamless internet access in the whole city by using the given infrastructure. These companies offer an accounting model for all private users who share their internet connectivity, so that the companies' customers may use these internet links. The task is, to provide a network protocol that authenticates the companies' customers to the companies and offers fair accounting for the private users, that share their internet connection with the customers.

Sastry et. al [13] made a new proposal for the network structure that is needed for realizing a city-wide wireless LAN access. Shortly, they propose that a foreign network provider (in the following called \mathcal{F}) does only relay the traffic between the mobile node (called \mathcal{M}) and the home network (called \mathcal{H}) which then acts as a proxy server for all services the mobile node wants to access. The communication between the mobile node and the home network is protected by a confidential and authenticated tunnel to improve the security. The big advantage of this solution is that the risk for the misuse of the foreign network's internet link drops to zero, because all services (including internet access) are provided by the home network. The single purpose of the foreign network \mathcal{F} is to relay the tunnel data between the mobile node \mathcal{M} and the home network \mathcal{H} .

Nevertheless, Sastry et. al did not propose a concrete implementation for this solution.

Manulis et. al [7] extended this idea with a concrete secure authentication and key establishment protocol for three parties. This protocol accomplishes mutual authentication between \mathcal{M} , \mathcal{H} and \mathcal{F} , \mathcal{H} , which is necessary for the secure communication and can later be used for accounting purposes also. Their proposed protocol is not optimized for efficiency in terms of roaming.

We propose a new network protocol that is optimized for roaming, even when multimedia services like VoIP or video streaming are in use. This can be reached by improving the efficiency in comparison to the proposed protocol by Manulis et. al. Furthermore, we present a protocol for accounting purposes so that a commercial scenario can be realized easily.

2 Our Roaming and Accounting Solution

Before we can go over to present our new protocols EAWRT (Efficient Authenticated Wireless Roaming via Tunnels) and WRA (Wireless Roaming Accounting), we have to define the environment for both protocols formally.

That is defining the protocol participants and the used key material (section 2.1), the identifiers for instances and protocol sessions in section 2.2, our trust assumptions (i.e. who trusts who) in section 2.3, the adversary abilities (section 2.4) and last but not least the building blocks (cryptographic primitives and abbreviations) of our protocols in section 2.5.

After that we present our roaming protocol (EAWRT) in section 3 and the accounting protocol (WRA) in section 4, giving proof sketches for the correctness as well as the security of the protocols.

2.1 Protocol Participants and Key Material

The protocol participants are namely the mobile device \mathcal{M} , a foreign network \mathcal{F} and a home network \mathcal{H} . The user of the mobile device \mathcal{M} has got a service contract with a home network \mathcal{H} , which gives him access to several provided services by \mathcal{H} , wherever an appropriate network infrastructure is given. An appropriate network infrastructure is realized through the nodes of the foreign network \mathcal{F} , that provide on the one side wireless access for all \mathcal{M} and on the other side a fast link to the home network \mathcal{H} .

We assume, \mathcal{M} and \mathcal{H} are in possession of a common longterm key k_{MH} that is chosen with respect to the security parameter l .

For relaying data between \mathcal{M} and \mathcal{H} , the foreign network wants to get paid. Therefore there is another contract between each foreign network \mathcal{F} and home network \mathcal{H} . Because there may be a lot of different home networks and even more foreign networks, it is not efficient to provide a symmetric key between each foreign network and each home network.

For that reason, each foreign network \mathcal{F} and home network \mathcal{H} own a Diffie-Hellman public key pair $\{SK, PK\}$ which is chosen with regard to the security parameter l .

2.2 Instances and Protocol Sessions

The number of the mobile devices \mathcal{M} , foreign networks \mathcal{F} and also home networks \mathcal{H} may be very big, so it is likely that the same \mathcal{F} or \mathcal{H} (or even \mathcal{M}) are participants in several parallel protocol sessions. We want to extend this by saying that it is possible that there are different protocol sessions with the same \mathcal{M} , \mathcal{F} and \mathcal{H} . The number of parallel protocol sessions is denoted as q (later used in the security analysis).

We claim that there is an unlimited number of instances of \mathcal{M} , \mathcal{F} and \mathcal{H} , whereby denoting an instance as \mathcal{X}_s with $\mathcal{X} \in \{\mathcal{M}, \mathcal{F}, \mathcal{H}\}$ and $s \in \mathbf{N}$. Three instances \mathcal{M} , \mathcal{F} and \mathcal{H} are called partnered when they have the same session id $SID := H, AID_M, F, r_H, r_M, r_F$ whereby H, AID_M, F are the identifiers of \mathcal{H} , \mathcal{M} , \mathcal{F} and r_H, r_M, r_F are randomly chosen nonces of each participant.

An instance of \mathcal{H} , \mathcal{M} , \mathcal{F} in a protocol session calls ACCEPT or ABORT upon the decision if the protocol execution was successful in respect to the protocol aims.

2.3 Trust Assumptions

Before protocol execution, the mobile device \mathcal{M} and the home network \mathcal{H} share some credentials that allow them to do a mutual authentication, which is necessary for establishing a trusted communication tunnel. Since \mathcal{H} provides a service

for \mathcal{M} , both parties must have a contract with each other, including on the one hand credentials and on the other hand rules for accounting and usage.

The foreign network \mathcal{F} is responsible for the relay of the tunnel data between the mobile device \mathcal{M} and the home network \mathcal{H} . Mutual authentication between \mathcal{F} and \mathcal{H} is required, because the foreign network \mathcal{F} clearly wants to get paid for the forwarding service it provides and must therefore be aware of \mathcal{H} 's identity. Additionally the home network \mathcal{H} wants to be sure about \mathcal{F} 's identity to realize a fair payment. Furthermore sharing credentials between \mathcal{F} and \mathcal{H} to support the accounting process may be necessary.

The mobile device \mathcal{M} will be implicitly authenticated against the foreign network \mathcal{F} due to the fact that \mathcal{H} accepts in the protocol. The same applies for the foreign network \mathcal{F} against \mathcal{M} , because the mobile device \mathcal{M} is assured that \mathcal{H} would not have been accepted when the authentication between \mathcal{F} and \mathcal{H} had failed.

2.4 Adversarial Model

The adversary \mathcal{A} is modelled as a probabilistic polynomial time (PPT) machine and has full control over the communication and protocol invocations. \mathcal{A} is allowed to do the following queries:

- **Invoke**(\mathcal{X}, m). Upon this query, a new instance \mathcal{X}_s of $\mathcal{X} \in \{\mathcal{M}, \mathcal{F}, \mathcal{H}\}$ is created. Message m is sent to the new instance, whereby the answer is directed to the adversary \mathcal{A} .
- **Send**(\mathcal{X}_s, m). This query sends a message m to \mathcal{X}_s . When \mathcal{X}_s has completed processing m , the response is sent back to \mathcal{A} . With the help of this query, \mathcal{A} 's control over the communication channel is modeled, since \mathcal{A} is able to stay passive by honestly forwarding each message or to become active by modifying m or even injecting a new message.
- **Corrupt**(\mathcal{X}). As response to this query, \mathcal{A} gets the longterm key of \mathcal{X} . That is k_M for \mathcal{M} , SK_F for \mathcal{F} and $\{SK_H, k_M \forall \mathcal{M}\}$ for \mathcal{H} . When \mathcal{X} becomes corrupted, all instances \mathcal{X}_s of \mathcal{X} become corrupted too.
- **RevealKey**(\mathcal{X}_s). If \mathcal{X}_s has already accepted, the adversary \mathcal{A} gets the session key as response to this query. The session key between \mathcal{M} and \mathcal{H} is k_{MH} , whereas the session key between \mathcal{F} and \mathcal{H} is denoted as k_{FH} .
- **TestKey**(\mathcal{X}_s). The adversary may query **TestKey**() to an accepted instance of a session. The instance \mathcal{X}_s chooses a random bit b and answers with a random value on $b = 0$ and with the session key $\{k_{MH}, k_{FH}\}$ on $b = 1$.

2.5 Building Blocks

Now, we itemize the cryptographic primitives that are used by the proposed protocols EAWRT (Fig. 1) and WRA (Fig. 2).

- A *cryptographic hash function* that provides preimage, second preimage and collision resistance [11]. Hash: $\{0, 1\}^* \rightarrow \{0, 1\}^l$. By $\text{Succ}_{\text{Hash}}^{\text{preimage}}(l)$ we denote

the success probability for a PPT adversary to find a preimage for a given output $\in \{0, 1\}^l$ of the hash function. $\text{Succ}_{\text{Hash}}^{2\text{nd-preimage}}(l)$ denotes the success probability for a PPT adversary to find a second preimage $\in \{0, 1\}^*$ for a given preimage-hash pair $\in \langle \{0, 1\}^*, \{0, 1\}^l \rangle$.

- A *message authentication code* (MAC) that suffices the weak unforgeability against chosen message attacks (WUF-CMA) [4]. $\text{Succ}_{\text{MAC}}^{\text{wuf-cma}}(l)$ denotes the success probability over all PPT adversaries to find a MAC forgery under access to the MAC oracle. A MAC is verified with $\text{ver}_{\text{key}}(\text{value})$.
- A *pseudo random function* PRF: $\{0, 1\}^l \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ for key derivation. We denote the maximum advantage over all PPT adversaries (running within time l) in distinguishing the outputs of PRF from the outputs of a random oracle better than $\text{Pr}=\frac{1}{2}$ by $\text{Adv}_{\text{PRF}}^{\text{prf}}(l)$.
- A *symmetric encryption scheme with integrity protection* that suffices the indistinguishability property under adaptive chosen ciphertext attacks (IND-CCA2) [2]. We denote the advantage that an adversary is able to decrypt (dec) at least one bit without knowing the used key as $\text{Adv}_{\text{DEC}}^{\text{ind-cca2}}(l)$.

Furthermore, the symmetric encryption scheme satisfies weak unforgeability against chosen message attacks. The adversary’s success probability to encrypt (enc) without the right key and gaining a valid ciphertext is $\text{Succ}_{\text{ENC}}^{\text{wuf-cma}}(l)$.

- A static *diffie-hellman key agreement* over a finite cyclic group, where the decisional diffie hellman (DDH) problem is strong. By $\text{Adv}_{\text{DH}}^{\text{ddh}}(l)$ we denote the advantage over all PPT adversaries to recognize a valid DH tuple.
- A *digital signature scheme* that provides existential unforgeability under chosen message attacks (EUF-CMA). The signing operation is denoted by sig_{SK} , and the according verification operation by ver_{PK} . The maximum success probability over all PPT adversaries of finding a forgery is represented by $\text{Succ}_{\text{SIG/VER}}^{\text{euf-cma}}(l)$.
- A set of *database operations*: **lookup**(AID_M) searches for the given index AID_M and returns the corresponding identity (\mathcal{M}). **add**() inserts a new assignment: $AID_M \rightarrow \mathcal{M}$.
- A set of *verification functions*: **validate** and **verify**. **validate** checks, if a value is within a logical range. The range may be of length one (an expected value). **verify** is used, when the expected value must be cryptographically computed, e.g. when the expected value must be hashed.

3 Roaming Protocol (EAWRT)

In the following, we propose a new protocol for the wireless roaming via tunnels scenario. We introduce a more efficient protocol than Manulis et al. by abandoning on digital signatures and asymmetric encryption. Due to this, we have smaller messages and we need less computation time. Additionally we support anonymity of the mobile device.

The EAWRT protocol is shown in Figure 1. \mathcal{M} , \mathcal{F} , \mathcal{H} are the identities of the participants and AID_M is the anonymous identity of \mathcal{M} .

$SK_i = i, PK_i = g^i \pmod p$ are the private respectively public diffie-hellman parameter for $i \in \{F, H\}$. In detail (but not shown in the figure), there is also a big prime p that conforms to the security level l and a base g that generates \mathbb{Z}_p^* .

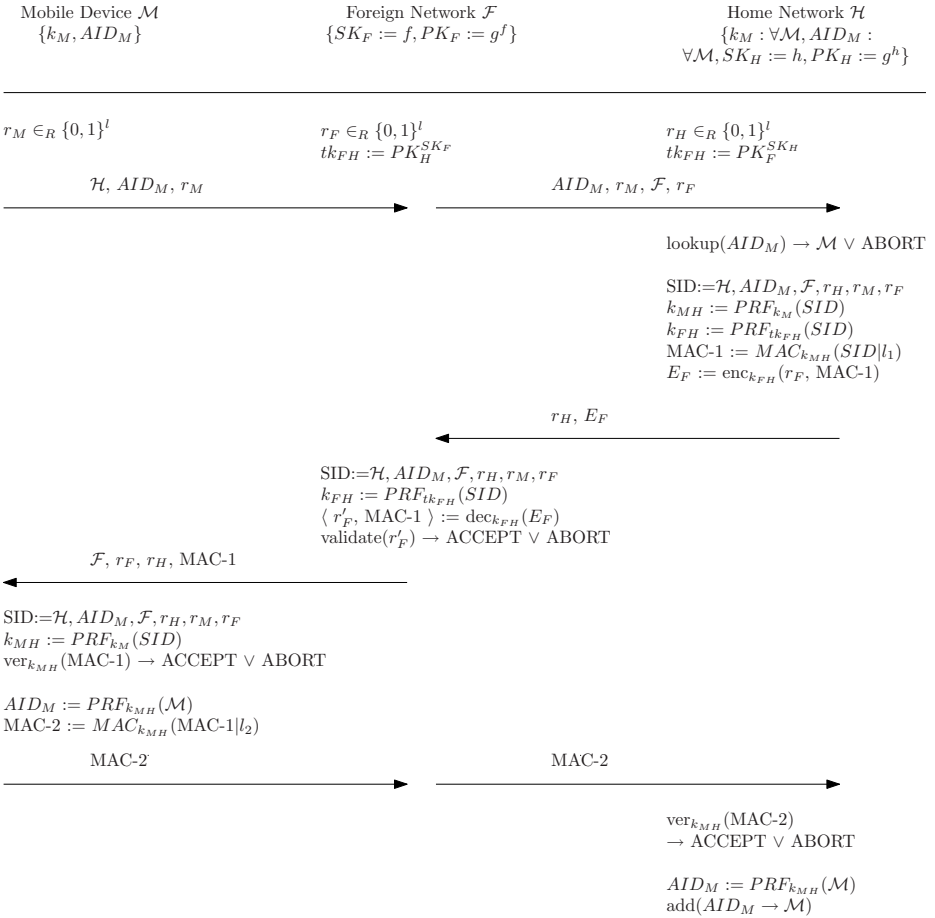


Fig. 1. Efficient Authenticated Wireless Roaming via Tunnels (EAWRT)

3.1 Correctness of EAWRT

The authentication and key establishment protocol Π (Figure 1) is correct, when definition 1 holds.

Definition 1 (Correctness EAWRT). *In the presence of a passive adversary, Π is correct when all parties \mathcal{M}, \mathcal{F} and \mathcal{H} have accepted and the key k_{MH} between \mathcal{M} and \mathcal{H} , as well as the key k_{FH} between \mathcal{F} and \mathcal{H} is identical on both sides.*

Proof (sketch). The key k_{MH} is computed as $PRF_{k_M}(SID)$, whereby k_M is a shared key between \mathcal{M} , \mathcal{H} and SID is the session identifier (consisting of all participant identifiers and all participant nonces). As proof statement we state that k_{MH} is identical on both sides, if both parties are partnered in the protocol session and share the same key k_M .

The key k_{FH} is computed as $PRF_{tk_{FH}}(SID)$, whereby tk_{FH} is a static Diffie-Hellman key between \mathcal{F} , \mathcal{H} and SID is the session identifier. If both instances are partnered in the protocol session, the public key of the other party is known and $PK_H^{SK_F} \equiv PK_F^{SK_H}$, then k_{FH} is identical on both sides.

The combination of both statements gives an idea for the correctness proof of EAWRT.

3.2 Security Definitions for EAWRT

Now we state the security goals that have to be achieved between the mobile device \mathcal{M} , the foreign network \mathcal{F} and the home network \mathcal{H} . Between \mathcal{M} and \mathcal{H} mutual authentication, integrity and confidentiality is required. These goals can be obtained by using symmetric cryptographic methods based on key material which is agreed on both sides. Non-repudiation is not explicitly required, which leads to the fact that no asymmetric cryptography is necessary.

Definition 2 (Mutual Authentication between \mathcal{M} and \mathcal{H}). *A wins if one of the following arises during the protocol run:*

1. An uncorrupted instance of \mathcal{M} accepts with a corrupted partnered instance of \mathcal{H}
2. An uncorrupted instance of \mathcal{H} accepts with a corrupted partnered instance of \mathcal{M}
3. After having accepted, both uncorrupted partnered instances \mathcal{M} and \mathcal{H} hold a different session key k_{MH} .

Definition 3 (Authenticated Key Exchange between \mathcal{M} and \mathcal{H}). *Given a uniformly chosen bit b , a PPT adversary \mathcal{A} interacts with a correct protocol Π , whereby it is not allowed for \mathcal{A} to query **RevealKey()** to an accepted instance or to corrupt an instance. $Game_{\Pi}^{ake-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l)$ is defined as the following interaction:*

1. \mathcal{A} interacts with instances of \mathcal{M} , \mathcal{F} , \mathcal{H} without using the **RevealKey()** and **Corrupt()** query
2. \mathcal{A} asks **TestKey()** to an instance of \mathcal{M} or \mathcal{H} and gets, dependent on b , a random value chosen from $\{0,1\}^l$ (if $b = 0$) or k_{MH} (if $b = 1$)
3. After further interaction, \mathcal{A} terminates and outputs a bit b'

\mathcal{A} wins $Game_{\Pi}^{ake-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l)$ if $b' = b$. The maximum probability of the adversarial advantage over the random guess of b , over all adversaries (running in time l) is

$$Adv_{\Pi}^{ake-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l) = \mathcal{A} \stackrel{max}{|} 2Pr[Game_{\Pi}^{ake-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l) = b] - 1|.$$

Definition 4 (Mutual Authentication between \mathcal{F} and \mathcal{H}). *A wins if one of the following arises during the protocol run:*

1. An uncorrupted instance of \mathcal{F} accepts with a corrupted partnered instance of \mathcal{H}
2. An uncorrupted instance of \mathcal{H} accepts with a corrupted partnered instance of \mathcal{F}
3. After having accepted, both uncorrupted partnered instances \mathcal{F} and \mathcal{H} hold a different session key k_{FH}

Definition 5 (Authenticated Key Exchange between \mathcal{F} and \mathcal{H}). *Given a uniformly chosen bit b , a PPT adversary \mathcal{A} interacts with a correct protocol Π , whereby it is not allowed for \mathcal{A} to query **RevealKey()** to an accepted instance or to corrupt an instance. $\text{Game}_{\Pi}^{\text{ake-}\mathcal{F}-\mathcal{H}}(\mathcal{A}, l)$ is defined as the following interaction:*

1. \mathcal{A} interacts with instances of \mathcal{M} , \mathcal{F} , \mathcal{H} without using the **RevealKey()** and **Corrupt()** query
2. \mathcal{A} asks **TestKey()** to an instance of \mathcal{F} or \mathcal{H} and gets, dependent on b , a random value chosen from $\{0,1\}^l$ (if $b = 0$) or k_{FH} (if $b = 1$)
3. After further interaction, \mathcal{A} terminates and outputs a bit b'

\mathcal{A} wins $\text{Game}_{\Pi}^{\text{ake-}\mathcal{F}-\mathcal{H}}(\mathcal{A}, l)$ if $b' = b$. The maximum probability of the adversarial advantage over the random guess of b , over all adversaries (running in time l) is

$$\text{Adv}_{\Pi}^{\text{ake-}\mathcal{F}-\mathcal{H}}(\mathcal{A}, l) = \max_{\mathcal{A}} |2\text{Pr}[\text{Game}_{\Pi}^{\text{ake-}\mathcal{F}-\mathcal{H}}(\mathcal{A}, l) = b] - 1|.$$

Definition 6 (Anonymity of \mathcal{M}). *This goal protects the anonymity of \mathcal{M} by hiding the real identity of \mathcal{M} towards \mathcal{F} and all protocol outsiders. A PPT adversary \mathcal{A} wins if one of the following occurs, after \mathcal{M} and \mathcal{H} have accepted:*

1. \mathcal{A} knows the real identity of \mathcal{M}
2. \mathcal{A} knows if an instance of \mathcal{M} has participated in a previous accepted session
3. \mathcal{A} recognizes an instance of \mathcal{M} when it participates in a next session

3.3 Proof Sketches for the Security of EAWRT

Having all definitions and assumptions in place, we proceed proving the security of the EAWRT protocol. To give a full security proof, we need to show that the protocol accomplishes the security goals from definition 2 to 6. Due to lack of space we only give short proof sketches and refer to [10] for the full security analysis of EAWRT.

Theorem 1 (Mutual authentication between \mathcal{M} and \mathcal{H}). *With a WUF-CMA secure MAC, the protocol Π of EAWRT provides mutual authentication in the sense of definition 2 and*

$$\text{Succ}_{\text{EAWRT}}^{\text{MA-}\mathcal{M}-\mathcal{H}}(l) \leq \frac{3q^2}{2^l} + 2\text{Succ}_{\text{MAC}}^{\text{wuf-cma}}(l).$$

Proof (sketch). The advantage of the adversary can be reduced to the collision of the three nonces r_M, r_F, r_H (within q sessions) and the adversary’s probability successfully forging MAC-1 or MAC-2.

Theorem 2 (Authenticated Key Exchange between \mathcal{M} and \mathcal{H}). *With a pseudo random function and a WUF-CMA secure MAC, the protocol Π of EAWRT provides authenticated key exchange in the sense of definition 3 and*

$$Succ_{EAWRT}^{AKE-\mathcal{M}-\mathcal{H}}(l) \leq \frac{3q^2}{2l} + 2Succ_{MAC}^{wuf-cma}(l) + 2qAdv_{PRF}^{prf}(l).$$

Proof (sketch). The advantage of the adversary can be reduced to the collision of the three nonces r_M, r_F, r_H , the adversary’s probability successfully forging MAC-1 or MAC-2 and the success probability of distinguishing k_{MH} from a random value in all q sessions.

Theorem 3 (Mutual Authentication between \mathcal{F} and \mathcal{H}). *With a WUF-CMA secure MAC, the protocol Π of EAWRT provides mutual authentication in the sense of definition 4 and*

$$Succ_{EAWRT}^{MA-\mathcal{F}-\mathcal{H}}(l) \leq \frac{3q^2}{2l} + Succ_{ENC}^{wuf-cma}(l) + qAdv_{DEC}^{ind-cca2}(l).$$

Proof (sketch). The advantage of the adversary can be reduced to the collision of the three nonces r_M, r_F, r_H and the adversary’s probability successfully forging an encryption E_F . Additionally, if the adversary is able to extract one bit of information out of E_F , it is possible to create a distinguisher that breaks the IND-CCA2 security. The advantage of breaking the IND-CCA2 security (q parallel sessions) is added to the success probability of the adversary.

Theorem 4 (Authenticated Key Exchange between \mathcal{F} and \mathcal{H}). *With a static Diffie-Hellman and a IND-CCA2 secure symmetric encryption, the protocol Π of EAWRT provides authenticated key exchange in the sense of definition 5 and*

$$Succ_{EAWRT}^{AKE-\mathcal{F}-\mathcal{H}}(l) \leq \frac{3q^2}{2l} + qAdv_{DH}^{ddh}(l) + qAdv_{DEC}^{ind-cca2}(l) + 2qAdv_{PRF}^{prf}(l).$$

Proof (sketch). The advantage of the adversary can be reduced to the collision of the three nonces r_M, r_F, r_H , the secrecy of tk_{FH} breaking the decisional Diffie-Hellman assumption, the security of E_F by being able to extract one bit of information about k_{FH} (IND-CCA2) and at last the pseudo randomness of k_{FH} due to being able to predict a random value for the used pseudo random function (PRF).

Theorem 5 (Anonymity of \mathcal{M}). *With a pseudo random function PRF, the protocol Π of EAWRT provides anonymity of \mathcal{M} in the sense of definition 6 and*

$$Succ_{EAWRT}^{anonymity}(l) \leq \frac{3q^2}{2l} + qAdv_{PRF}^{prf}(l).$$

Proof (sketch). The advantage of the adversary can be reduced to the collision of the three nonces r_M, r_F, r_H and the pseudo randomness of AID_M .

Combining the previous five theorems, we state security of EAWRT according to the defined security goals.

4 Accounting Protocol (WRA)

We extended the model of Manulis et al. by the need for a fair accounting. To realize that, we propose the WRA protocol, which is an extension to the normal tunnel communication between the mobile device \mathcal{M} and the home network \mathcal{H} . Additionally to the tunnel data, which is represented by MSG and MSG2, we have added some cryptographic measures to ensure that the foreign network \mathcal{F} is able to proof, how many data was relayed. As consequence, the foreign network \mathcal{F} is able to bring this size of transmitted data to account, whereby neither \mathcal{H} nor \mathcal{F} is able to cheat.

The home network \mathcal{H} acknowledges the size of the transmitted data to F via two mechanisms. Firstly as an absolute value of the transmitted bytes in a

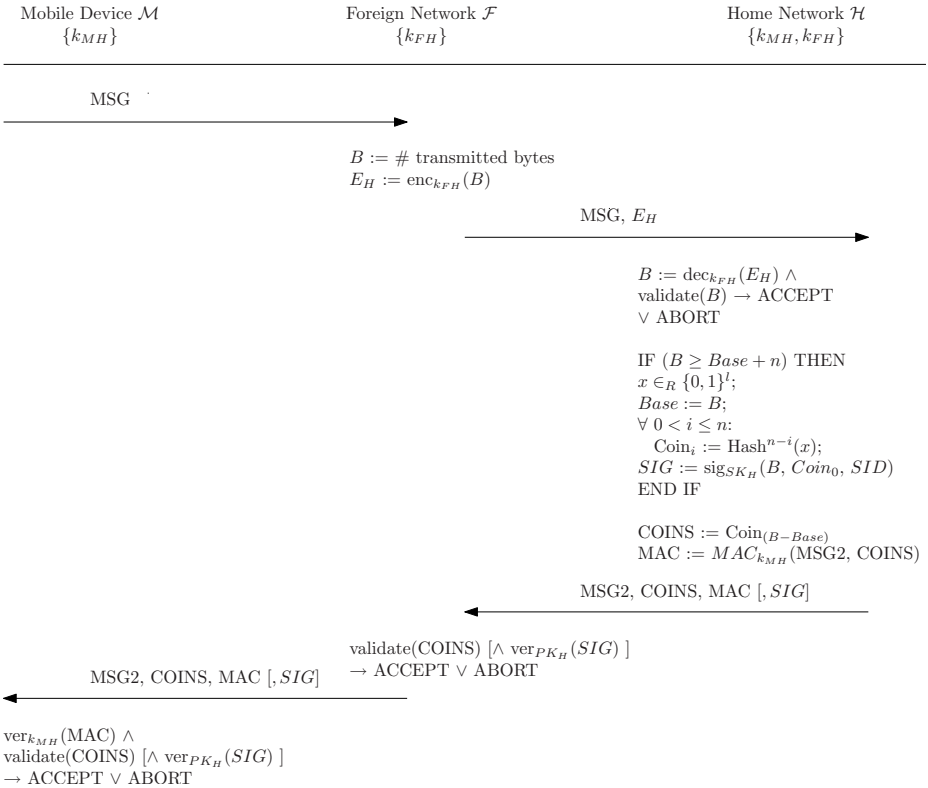


Fig. 2. Wireless Roaming Accounting protocol (WRA)

digital signature. Secondly as an element of a hash chain, representing a value relative to the last digitally signed value.

Figure 2 shows the WRA protocol. The size of the used hash chain is denoted by n . B is the number of transmitted bytes, whereby $Base$ is the last digitally signed value of B .

4.1 Correctness of WRA

The accounting protocol is correct when definition 7 holds.

Definition 7 (Correctness WRA). *In the presence of a passive adversary, Π is correct when \mathcal{M} , \mathcal{F} and \mathcal{H} have accepted and are sure that the partnered instances hold the same value B , containing the transmitted data volume.*

Proof (sketch). We give an idea for the correctness proof of WRA in the following. If all parties have accepted, it is left to show that all parties have the knowledge of the same value B in the presence of a passive adversary. B can be represented by several values: B , COINS and SIG . \mathcal{H} sends COINS with a corresponding MAC in the third message, \mathcal{F} forwards these values in the fourth message. If these values have arrived at \mathcal{M} and \mathcal{M} accepts, it is obvious that all parties hold the same value for B .

The correctness of WRA can be proven with these considerations.

4.2 Security Definitions for WRA

Between \mathcal{F} and \mathcal{H} mutual authentication is required for accounting. Both sides have to be sure about the identity of the other party, so that one side can account its provided service and the other side will accept the issued bill. Integrity protection and maybe confidentiality are necessary to protect the accounting data communicated between \mathcal{F} and \mathcal{H} .

Definition 8 (Fair Accountability). *In order to guarantee fair accountability, the foreign network \mathcal{F} needs a non-repudiative acknowledgement over the size of the data, that was forwarded.*

By demonstrating this acknowledgement, the foreign network \mathcal{F} can prove, how much data was relayed (at least), whereby nor the mobile device \mathcal{M} neither the home network \mathcal{H} are able to deny this. \mathcal{A} wins if one of the following arises during the protocol run:

1. *An uncorrupted instance of \mathcal{F} or \mathcal{M} accepts an acknowledgement over the transmitted bytes (COINS/SIG) that was not created by \mathcal{H}*
2. *An uncorrupted instance of \mathcal{F} or \mathcal{M} accepts an invalid or replayed acknowledgement over the transmitted bytes (COINS/SIG)*

4.3 Proof Sketch for the Security of WRA

The WRA protocol fulfills the fair accountability property according to definition 8. Due to lack of space we only present a proof sketch here. The full proof of this theorem can be found in [10].

Theorem 6 (Fair Accountability of WRA). *Given a EUF-CMA secure digital signature scheme and a cryptographic hash function, the fair accountability property of WRA (definition 8) can be broken with a probability of*

$$\text{Succ}_{\text{WRA}}^{\text{FA}}(l) \leq \frac{1}{m} \text{Succ}_{\text{SIG/VER}}^{\text{euf-cma}}(l) + n \text{Succ}_{\text{Hash}}^{\text{preimage}}(l) + \text{Succ}_{\text{MAC}}^{\text{wuf-cma}}(l).$$

Proof (sketch). Given the appearance probability of SIG as $\text{Pr}[\text{SIG occurs}] := \frac{1}{m}$ with $1 < m \leq n$ and n the length of the used hash chains. The adversary's advantage can be reduced to forging (EUF-CMA) a signature SIG, appearing with the probability $\frac{1}{m}$, the forgery of COINS to a higher value by finding any valid preimage (maximum n) for a given value and the success probability of forging the value MAC (WUF-CMA).

5 Efficiency of EAWRT and WRA

In comparison with the WRT protocol from Manulis et al. [7], the EAWRT protocol has some obvious advantages in respect to performance, since we abandon digital signatures and asymmetric encryption. Due to this, we have notably smaller sized messages and less computation time needed. Particular for mobile devices this approach fits good, because their computation power respectively battery power is limited.

Moreover, we are able to improve the performance from EAWRT even more by applying some precomputations. The computation of tk_{FH} , the static diffie-hellman key, is computational expensive but has to be done only one time for all protocol instances with the same \mathcal{F} and \mathcal{H} . So, this key can be computed at the first contact between \mathcal{F} and \mathcal{H} and then stored for later use.

After the last message of the EAWRT protocol, \mathcal{H} verifies MAC-2 by comparison with a self-computed MAC-2. This computation can be done earlier to save time. The verification MAC-2 can be computed by \mathcal{H} right after sending out his message $\langle r_H, E_F \rangle$, while waiting for the last message of the protocol.

Our accounting protocol (WRA) works like an add-on to the normal communication (MSG, MSG2). To each message, only one hash value (COINS) and one MAC value is added (in total 256-512 bits). Additionally, everytime our hash chain runs out of preimages, a digital signature setting a new root for the hash chain is appended.

Since digital signatures need a big amount of space (usually 1024-4096 bits without canonicalization) and computation time, it is desirable to abandon them as long as possible. Therefore the length of the hash chain n should be chosen in respect to efficiency.

6 Conclusion

In this paper, we introduced two new properties for the wireless roaming via tunnels scenario. At first, the anonymity property, which allows the user of the mobile device to stay anonymous for outsiders (including the foreign network) while roaming. This includes the unlinkability of two different sessions.

The second property is named fair accounting, which has a special meaning for this scenario. It is necessary for the foreign network, which forwards the tunnel data between the mobile device and the home network, that the home network approves the size of the transmitted data. Since the foreign network wants to get paid for relaying, the home network's confirmation of the size of the transmitted data must be non-repudiative, in other words: signed. In dispute, the foreign network can present the signatures and demand the payment.

We have presented an optimized AWRT protocol (named EAWRT), that fulfills the requirements proposed by Manulis et al. [7]. Additionally, our protocol has the anonymity property *and* is designed to be more efficient. Notably the efficiency of our protocol is important, since we want to allow near realtime services like VoIP or video chats even in roaming cases.

Moreover we showed up a solution for the accounting problem by introducing another protocol named WRA. This protocol attaches some cryptographic values to the normal communication flow and can thereby enforce fair accounting without too much overhead.

For both introduced protocols there is a full security analysis in [10].

References

1. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowitz, H.: Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard), Updated by RFC 5247 (June 2004)
2. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
3. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing* 17, 281–308 (1988)
4. Bellare, M., Namprempe, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
5. Bellare, M., Kilian, J., Rogaway, P.: Security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences* 61(3), 362–399 (2000)
6. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication
7. Manulis, M., Leroy, D., Koeune, F., Bonaventure, O., Quisquater, J.-J.: Authenticated wireless roaming via tunnels: Making mobile guests feel at home. *Cryptology ePrint Archive*, Report 2008/382 (2008), <http://eprint.iacr.org/>
8. Manulis, M., Sadeghi, A.-R., Schwenk, J.: Linkable democratic group signatures
9. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)

10. Noack, A.: Efficient authenticated wireless roaming via tunnels. Technical Report, Ruhr-University Bochum (2009), http://nds.hgi.rub.de/noack/No_EAWRT_full.pdf
11. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 371–388. Springer, Heidelberg (2004)
12. Pointcheval, D., Stern, J.: Provably secure blind signature schemes, pp. 252–265. Springer, Heidelberg (1996)
13. Sastry, N., Sollins, K., Crowcroft, J.: Architecting citywide ubiquitous wi-fi access. In: HotNets-VI (2007), <http://conferences.sigcomm.org/hotnets/2007/papers/hotnets6-final88.pdf>
14. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), <http://eprint.iacr.org/>

AAA-IDEA 2009

Session II – SOA and Web Systems

Towards the Integration of Distributed Transactional Memories in Application Servers' Clusters^{*}

Paolo Romano, Nuno Carvalho, Maria Couceiro,
Luís Rodrigues, and João Cachopo

INESC-ID, Lisbon, Portugal

Abstract. The transition to multicore architectures has raised the urge to identify novel programming paradigms aimed at simplifying the development of parallel programs.

Transactional Memories (TM) are regarded as one of the most promising approaches to address this issue, as highlighted by the huge interest garnered in the research community over the last years. Distributed Transactional Memories (DTMs) represent a very recent branching of the research line on TMs, aimed at enhancing their scalability and dependability.

In this paper, we review some of our recent results and research directions focused on the integration of DTMs in clusters of web application servers and on the design of scalable and fault-tolerant DTM algorithms.

1 Introduction

Transactional Memories (TMs) have garnered considerable interest of late due to the recent technological trend that has made of multi-core and many-core CPUs the architecture-of-choice for mainstream computing. TMs represent an attractive solution to spare programmers from the pitfalls of conventional explicit lock-based thread synchronization, relying instead on proven concurrency-control concepts used for decades by the database community to simplify concurrent programming [1]. When using TMs, the programmers are simply required to specify which operations on shared data structures are to be executed within the scope of an atomic and isolated transaction. By relinquishing the programmer from the burden of managing locks or other error-prone low-level concurrency control mechanisms, TMs have been shown to enable a sensible boost in productivity, as well as in code reliability, e.g., [6].

Even though the study of TMs has garnered a large interest in the research community over the last 5 years, the problem of how to enhance their scalability and fault-tolerance via distribution and replication has started to receive attention only very recently [2, 4, [12, 23]]. This is actually a major gap, which becomes pretty manifest when TMs start to be adopted in real world applications, as they

^{*} This paper was partially supported by the Pastramy (PTDC/EIA/72405/2006) projects.

are faced with harsh scalability and dependability challenges that cannot be effectively tackled, due to the current lack of efficient Distributed Transactional Memory (DTM) solutions.

Web applications represent an important class of the systems that would significantly benefit from the adoption of TM-based solutions, provided that these are able to ensure adequate levels of scalability and failure resilience. Modern Web-based applications, in fact, tend to be structured according to a three-tier (or, more in general, multi-tier) architecture that relies on relational DBMSs for persisting the application data, whilst exploiting object-oriented programming platforms (e.g., J2EE) hosted by dedicated application servers for implementing the business logic. This allows reflecting at both the software and hardware level the logical decomposition of applications, permitting to achieve high modularity and flexibility. On the other hand, the partitioning of the application into multiple tiers generates an obvious increase in the system's complexity, generating performance and reliability pitfalls and hindering developers' productivity. Accessing the data on remote DBMSs, in fact, imposes incurring into onerous round-trips that may significantly hamper performance, especially for the case of complex operations. Further, the multiplicity and diversity of the employed components, and their interdependencies, makes reliability a complex issue to tackle, exposing the system to a spectrum of hazardous state inconsistencies in the presence of failures [14, 32]. Finally, rather than being a completely transparent aspect, relational-based persistence affects the programming model, hindering the successful implementation of an object-oriented rich domain model.

To overcome these problems, in [6, 9], we introduced a novel, TM-centric approach to architect web applications. In such an approach, the application's state is hosted in memory by the application servers, and locally persisted for scalability and durability purposes. The increased locality between the application logic and data alleviates the aforementioned performance and reliability issues. Accesses to the application state are handled, in a totally transparent manner for the developers, by a DTM layer that (i) enforces the atomicity and the isolation of any state updates triggered by the application, (ii) guarantees the consistency of the application state replicated across the nodes of the cluster, and (iii) triggers, when necessary, the update of a lightweight persistent storage system that is also replicated across the cluster.

In this paper we describe some of our recent results towards the realization of the envisioned TM-centric architecture, focusing in particular on the issues related to the design and implementation of scalable and dependable DTMs (rather than, e.g., on the aspects related to persistence).

We start by reporting our experiences with the development of the FénixEDU application, which represents, to the best of our knowledge, the first web application to have leveraged on TM technology. Next, we report the results of a workload characterization study of FénixEDU, whose results have driven some of our main choices in the design space of the algorithms architected to ensure the consistency of DTM platforms. We then describe BFC (Bloom Filter Certification), a novel TM replication protocol, which we recently proposed in [12], that exploits an

efficient Bloom Filter-based encoding technique to reduce the overhead associated with the cluster-wide certification of transactions. Finally, we point out some of our current research directions in this area.

This paper is organized as follows. Section 2 presents related work. Section 3 introduces the FénixEDU system, describing its current architecture and highlighting some key aspects of its workload. In Section 4 we illustrate the proposed architecture. The BFC replication protocol is overviewed in Section 5. Section 6 concludes the work and points to future research directions.

2 Related Work

One way to implement a web application domain model is to use an object-oriented paradigm. A common approach is, for instance, to use a multi-tier J2EE architecture where the business logic and data are modeled using Enterprise Java Beans, being the data stored in databases by means of an Object/Relational mapping tool. The solution proposed in [29] presents a way to replicate such systems by adding fault tolerance mechanisms on both the application server and the database. The authors rely on a locking based approach. In our case, by relying on a DTM to synchronize concurrent accesses directly at the application server's tier, we are able to avoid error-prone, explicit locking schemes, and to rely on much simpler, and more lightweight, persistence solutions rather than on fully-fledged relational databases.

The only DTM solutions that we are aware of are those in [2, 4, 23]. However, the solutions proposed so far have not addressed the important issue of how to exploit replication not only to improve performance, but also to enhance dependability. This is clearly a central aspect of DTM's design, as the probability of failures increases with the number of nodes, becoming impossible to ignore in large clusters.

The problem of replicating a TM is closely related to the problem of database replication, given that both TMs and DBMSs share the same key abstraction of atomic transactions. The fulcrum of modern database replication schemes [27, 28] is the reliance on an Atomic Broadcast (ABcast) primitive [13, 19], typically provided by some Group Communication System (GCS) [25]. Roughly speaking, an ABcast service ensures that the broadcast messages are received by all or none of the participants, and in the same order, despite the occurrence of failures. ABcast plays a key role to enforce, in a non-blocking manner, a global transaction serialization order without incurring in the scalability problems affecting classical eager replication mechanisms based on distributed locking and atomic commit protocols, which require much finer grained coordination and fall prey of deadlocks [16]. Certification-based approaches, such as [28, 29, 30] are considered as some of the most efficient solutions among the plethora of ABcast based replication schemes [12]. Therefore, they represent natural candidates also in the context of TM systems. In these schemes, transactions are locally processed on a single replica and validated *a posteriori* of their execution through an ABcast based certification procedure aimed at detecting remote conflicts between concurrent transactions. The certification based approaches may be further classified

into voting and non-voting schemes, where voting schemes, unlike non-voting ones, need to ABcast only the writeset (and not the readset which may be very large), but on the other hand incur in the overhead of an additional uniform broadcast [19] along the critical path of the commit phase. As highlighted in our previous work [31], the replica coordination latency has a significantly amplified cost in TMs when compared to conventional database environments, given that the average transaction execution time in TM settings is typically several orders of magnitude shorter than in database ones. To maximize efficiency, it is therefore highly desirable to design novel mechanisms capable of minimizing the costs associated with the replica coordination schemes. This represents an important goal of our current research activities.

Our work is also related to the large body of literature on Distributed Shared Memories (DSM). To overcome the strong performance overheads introduced by classic DSM implementations [24, 35], which ensure strong consistency guarantees with the granularity of a single memory access, several DSM systems provide relaxed memory consistency guarantees, e.g., [21]. Unfortunately, developing software for relaxed DSM's consistency models may be challenging for programmers because they need to master complicated consistency properties. Conversely, the simplicity of the atomic transaction abstraction, at the core of (D)TMs, allows to increase programmers' productivity [6] with respect to both locking disciplines and relaxed memory consistency models. Further, the strong consistency guarantees provided by atomic transactions can be supported through efficient algorithms that incur only in a single synchronization phase per transaction (typically taking place at commit time), effectively amortizing the communication overheads across a set of (possibly large) memory accesses.

3 The FénixEDU System

The FénixEDU system is a web application that supports a wide range of academic activities in the Lisbon's Instituto Superior Técnico (IST) campus (management of web pages for different courses, student enrollment, etc). The FénixEDU system started as a typical web application, with the application logic implemented in Java and hosted by a single application server, and its data stored in a relational DBMS. In its first version, FénixEDU relied on an Object/Relational mapping tool [26] to store the objects in the database, while maintaining a local cache of the database data. To control the concurrent access to the domain entities, FénixEDU relied on explicit lock-based interfaces to synchronize read and write operations [11]. Unfortunately, this lock-based approach to concurrency was highly error-prone, as programmers often forgot or misplaced the acquisition of locks, causing frequent consistency problems into the domain data. Moreover, with the increased usage of the system, also the first performance problems appeared. After some performance profiling, these were attributed to the overheads incurred in the acquisition and in the management of locks by the operations that accessed many thousands of objects.

To address these issues, across 2005 the FénixEDU codebase was adapted to permit transparent integration with a TM layer, called JVSTM [7]. JVSTM

relies on a software based implementation of a multiversion concurrency control scheme [3], providing excellent performance for read-only transactions (largely predominating in reference benchmarks for Web applications [36, 37], as well as in the FénixEDU's workload), because they incur in negligible book-keeping overheads and are sheltered from the possibility of blocks or aborts. The integration of JVSTM within the architecture of the FénixEDU application was designed so to achieve total transparency from the developer's perspective, and provided benefits not only in terms of performance (thanks to the elimination of the overheads associated with lock acquisition and management), but also in terms of robustness (thanks to the avoidance of the error-prone manual management of locks) and simplification of the programming model (quantifiable in terms of reduction of lines of code to be developed and debugged [6]).

Serving a population of 12000 students, 900 faculty and 800 administrative members and faced with a steadily increasing traffic volume, the FénixEDU system was eventually forced to address the problem of scaling out the TM-enabled application server. As a first step in this direction, a very simple replica synchronization scheme orchestrated by a centralized back-end database is currently being employed. Essentially, each application server is required to access the database every time it starts a new transaction (whether read-only or not) to check whether its local cache is still up-to-date. The detection of any conflict developed during transactions' execution is performed at commit time via a sequential validation phase performed by checking whether the readset of the committing transaction T intersects with the writesets (stored by the database) of any transaction that has committed before T. Unfortunately, even though this approach is very simple, the reliance of the current replication solution on an external data storage causes large performance overheads, strongly limits concurrency, and suffers of a single point of failure.

To drive the design phase of an efficient DTM platform capable of effectively matching the characteristics of the FénixEDU application, the system was instrumented to collect information on the nature of the workload generated by the application towards the TM layer. The workload data was collected over a period of two years (from June 2007 to July 2009), gathering information concerning around 390 millions of transactions. A first result that we observed is that write transactions are approximately only 2% of the total number of transactions in the system. This ratio supports the choice of a TM layer, such as JVSTM, particularly optimized for read-only transactions. Further, it suggests to bias the design of a DTM platform so to require synchronization exclusively for write transactions, unlike the currently operational solution that demands a remote synchronization with the back-end also when a read-only transaction is started.

In Figure 1 we plot the probability density functions of the readset's and writeset's sizes for write transactions, as these factors may have a big influence on the amount of information exchanged among the replicas of a DTM. From these plots we can draw two main considerations. On one side, we observe that, on average, writesets are several orders of magnitude smaller than readsets (more precisely the average readset's size is of 5575, whereas the average writeset's size

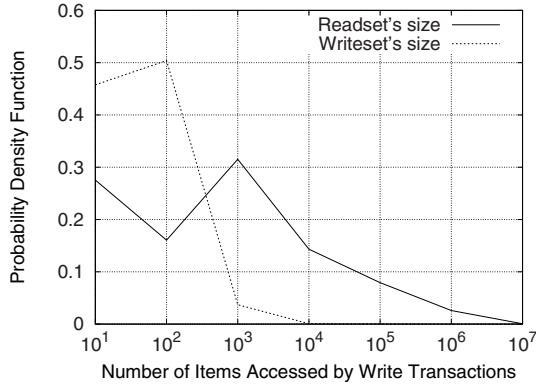


Fig. 1. Probability density functions for the readset’s and writeset’s size of write transactions

is of 36). This suggests that DTM solutions should strive to avoid communicating the whole readset and, whether possible, should exclusively propagate information concerning the transactions’ writeset. On the other hand, the sizes of the readset and the writeset are far from being concentrated around their average values: the maximum readset’s and writeset’s sizes, in fact, are from three to four orders of magnitude larger than the corresponding average values. In other words, the FénixEDU’s workload comprehends very heterogeneous components, which makes it extremely challenging to identify a “one-size-fits-all-solution” capable to deliver optimal performances in every scenario.

4 System Architecture

To address the above discussed inefficiencies and limitations affecting the DTM solution currently supporting the FénixEDU application, in [9] we have recently proposed a new architecture that neatly decouples the issues related with the synchronization of the replicated TM layer with those concerning the persistence of data. In the envisioned architecture, which is depicted also in Figure 2, the consistency of the TM-enabled application server replicas is no longer dependent on a centralized DBMS. Conversely, the application server replicas coordinate the execution of transactions by directly communicating among them, leveraging on the services provided by a Group Communication System to propagate the updates and reach cluster-wide agreement on the outcome (commit/abort) of transactions.

Coping with the issues related to the concurrent execution of distributed transactions directly at the application server replicas’ level, rather than relying on the assistance of a standard relational DBMS, provides two main advantages.

First, the achievement of a neater separation of logical concerns: once the TM tier has autonomously ensured the consistency of a transaction, the back-end just has to be able to atomically persist the corresponding updates. This

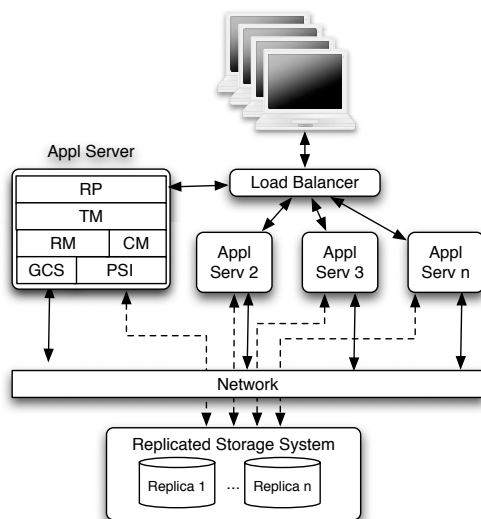


Fig. 2. The proposed DTM based architecture

permits to rely on much simpler, and more lightweight, persistence solutions than fully-fledged relational databases (which incur in the unnecessary overheads associated with SQL or complex concurrency control mechanisms [34]).

Further, the reliance on a standard, relational DBMS as the coordinator of the TM replication protocol forces to implement the whole replication protocol by exploiting exclusively standard SQL interface. Being SQL designed for other purposes, it can significantly hinder the development of even basic mechanisms typically employed by any transactional replication scheme (e.g., synchronous propagation of state updates to other replicas).

Let us now analyze more in detail the architecture illustrated in Figure 2. The incoming requests are dispatched by a load-balancer (see [8] for a comprehensive survey on load-balancing in web clusters) to a set of replicated application clusters, which rely on a replicated persistent storage for ensuring durability. Note that the latter is depicted as a logical independent component, even though it could be physically colocated in the same machines also hosting the application server to enhance locality between the application logic and the (persistent) data. In the remainder, we will concentrate on the description of the modules composing the DTM layer, which represents the actual focus of this paper, postponing a thorough analysis of the replicated persistence storage to a future paper.

Each application server hosts the following components: a *Request Processor* (RP), responsible for receiving the requests and activating the transactional logic; a *TM* instance, extended with a reflective interface that externalizes key information about the transactions' execution state (e.g., transactions' readsets and writesets), which are normally encapsulated by existing TM implementations; a *Cache Manager* (CM), responsible for implementing caching policies (e.g., prefetching, eviction strategies) based on the application access patterns;

a *Replication Manager* (RM), implementing the distributed coordination protocol required for ensuring replica consistency (an overview of the TM replication protocols currently under development/evaluation will be provided in the remainder of this paper); a *Persistent Store Interface* (PSI), providing APIs to interact with a replicated storage system; a *Group Communication Service* (GCS), responsible for maintaining up-to-date information regarding the membership of the group of application servers (including failure detection) and providing the required communication support for the coordination among the servers.

5 The BFC Protocol

The Replication Manager, being in charge of ensuring the consistency of the DTM layer, represents a fundamental building block of the architecture described in Section 4. In this section we present one of our recent results concerning the design and evaluation of TM replication mechanisms, namely the Bloom Filter Certification (BFC) protocol [12].

As already discussed in Section 2, the abundant literature on database replication protocols, and in particular the recently proposed family of AB-cast based replication schemes [28, 29, 30], represents a natural source of inspiration for the design of TM replication solutions. However, the efficiency of any transactional replication scheme is much more strained in TMs than in databases, given that the average transaction's execution time in TMs is typically several orders of magnitude smaller than in databases (in [31], for instance, we've shown that 50% of write transactions complete in less than $200\mu\text{secs}$ when considering standard TM benchmarks). This translates into a corresponding increase of the overhead associated with the inter-replica coordination activities, urging for novel solutions aimed at minimizing such costs.

The BFC protocol aims at achieving exactly this goal, requiring just a single ABcast to commit a transaction, like in non-voting certification protocols and differently from voting ones, which incur in the costs of an additional Uniform Reliable Broadcast. On the other hand, analogously to voting certification protocols, and unlike non-voting ones, BFC avoids to flood the network with large messages carrying the whole transaction's readset. The latter aspect is particularly important given that it is well-known that the ABcast latency is significantly affected by the size of transmitted messages [18, 20] and that the transactions' readsets are frequently very large in web applications. This is also confirmed by the results of the workload characterization study of the FénixEDU application reported in Section 3.

BFC achieves such a result by exploiting the space-efficient encoding properties of Bloom Filters (BF), whose fundamentals we briefly recall in the following for the sake of clarity (the interested reader may refer to [5] for a recent survey on BF and on their applications). BFs are data structures that permit to test whether an element is a member of a set, avoiding the encoding of the whole set, but rather permitting to store a much more compact representation of it. This comes, however, at the cost of incurring in false positives (i.e., an element may

appear to be present in the set, whereas it is not), albeit false negatives are, on the other hand, not possible. More in detail, a Bloom Filter representing a set $S = \{x_1, x_2, \dots, x_n\}$ of n elements from a universe U consists of an array of m bits, initially all set to 0. The filter uses k independent hash functions h_1, \dots, h_k with range $\{1, \dots, m\}$, which map each element in the universe to a random number uniformly over the range. To add an element $x \in S$ to a BF, x is fed to each of the k hash functions. The array positions output by the k hash functions are then all set to 1. To determine whether an item y belongs in S , the values of the $h_i(y)$ bits are checked. If even only one of these bits is 0, it means that y is not a member of S (with no possibility of mistakes). If all $h_i(x)$ are set to 1, then x may be in S , although this may be wrong with some probability, called *false positive* probability. Interestingly, the probability of a false positive f for a single query to a Bloom Filter can be known beforehand, once the number of bits used per item m/n and the number of hash functions k are fixed, by using the following formula:

$$f = (1 - e^{-kn/m})^k \quad (1)$$

We may now start describing the BFC scheme. Similarly to existing certification-based transactional replication schemes, in BFC incoming transactions are locally processed in an optimistic fashion, avoiding any inter-replica synchronization scheme during transaction execution. Further, by leveraging on the JVSTM multi-version scheme, the BFC ensures that read-only transactions are always provided with a consistent committed snapshot. This spares them from the risk of aborts and permits to obviate the need for replica coordinations even during the commit phase. Overall, the overhead incurred in by read-only transactions due to the replication scheme is in practice almost nullified.

For what concerns update transactions, at commit time these are first locally validated to detect any local conflicts. If the local validation phase is successfully passed, the Replication Manager encodes the transaction's readset (i.e., the set of identifiers of all the objects read by the transaction) in a BF, and ABcasts it along with the transaction writeset.

As in classical non-voting certification protocols, update transactions are validated by all replicas once they are ABcast-delivered. At this stage, replicas check whether the BF of the validating transaction contains any item updated by any concurrent transactions. If no match is found, given that a BF provides no false negatives, then the transaction may be safely committed. Otherwise the transaction is aborted.

Given that the occurrence of false positives leads to the generation of unnecessary aborts, the BF size is set so to ensure that the probability p_{abort} for a false positive to induce a transaction abort is bounded by a user-tunable threshold, which we denote as *maxAbortRate*. The problem here is that p_{abort} is a function of the number of queries q that will be performed on the BF during the transaction's validation phase, but the BF has to be constructed when the transaction enters the commit phase. At this time, however, it is not possible to predict the value of q , which is determined by the number of transactions that will commit before the transaction is ABcast-delivered and by the size of the writesets of each of these

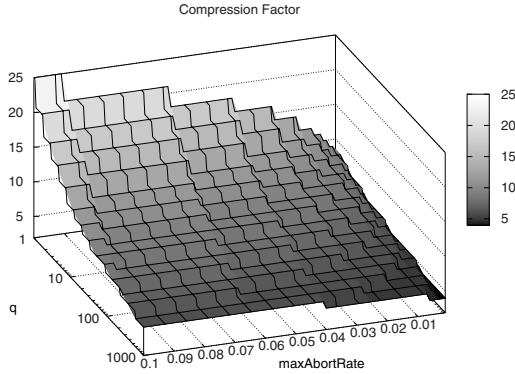


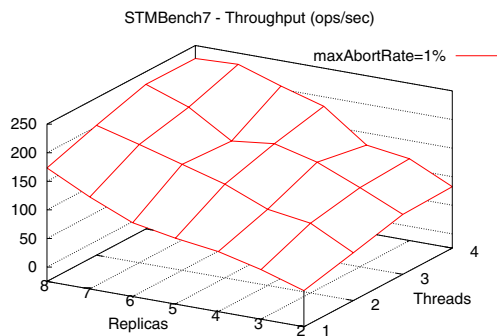
Fig. 3. Compression factor achieved by BFC considering the ISO/IEC 11578:1996 UUID encoding

transactions. Neither of these are known when the BF is created. On the other hand, it is important to highlight that any error in estimating q does not compromise consistency, but may only lead to deviations from the target $maxAbortRate$ threshold. To tackle this problem, BFC uses a lightweight heuristic that estimates q through the moving average across the number of BF queries performed during the validation phase of the last c transactions to have been ABcast-delivered. Once q is estimated, we can then determine the number m of bits in the BF by considering that the false positives for any distinct query are independent and identically distributed events which generate a Bernoullian process where the probability of occurrence of a single event (namely, a false positive during a single query) is given by Equation 1. After some simple maths, we obtain the following expression for the BF’s size:

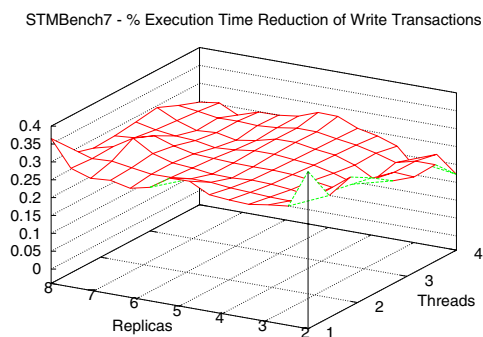
$$m = \left\lceil -n \frac{\log_2(1 - (1 - maxAbortRate)^{\frac{1}{q}})}{\ln 2} \right\rceil$$

The striking reduction of the amount of information exchanged, achievable by the BFC scheme, is clearly highlighted by the graph in Figure 3, which shows the BFC’s compression factor (defined as the ratio between the number of bits for encoding a transaction’s readset with the ISO/IEC 11578:1996 standard UUID encoding, and with BFC) as a function of the target $maxAbortRate$ parameter and of the number q of queries performed during the validation phase. The plotted data shows that, even for marginal increases of the transaction abort probability in the range of [1%-2%], BFC achieves a [5x-12x] compression factor, and that the compression factor extends up to 25x in the case of 10% probability of transaction aborts induced by a false positive of the Bloom Filter.

To evaluate the scalability of the BFC protocol, and quantify the performance gains achievable with respect to conventional certification based protocols, we developed a prototype implementation. Differently from the architecture described in Section 4, the current implementation of BFC is not yet interfaced with a



(a) Throughput



(b) % Execution Time Reduction

Fig. 4. STMBench7, read dominated with long traversals, $maxAbortRate=1\%$

persistent storage system and assumes that each replica maintains a whole copy of the DTM (hence not needing the Cache Manager component). Extending our current prototype to incorporate these modules represents an important direction for our future work. On the other hand, the current prototype permits us to evaluate empirically the performance of the BFC scheme without incurring in possible interferences generated by the coexistence of other components, such as the Persistent Storage and the Cache Manager, which address orthogonal issues with respect to BFC.

The target platform for our experimental performance study is a cluster of 8 nodes, each one equipped with an Intel QuadCore Q6600 at 2.40GHz equipped with 8 GB of RAM running Linux 2.6.27 and interconnected via a private Gigabit Ethernet. We use Appia [10, 25] as the GCS and select a classic sequencer-based implementation [13, 19] for the ABcast service.

We consider a standard, and rather complex, benchmark for TM systems, namely *STMBench7* [17], which features a number of operations with different levels of complexity over an object-graph with millions of objects. Specifically, we consider the “read dominated with long traversals” configuration of the benchmark, which generates a workload closely resembling the one of the *FénixEDU* application (large readsets, much smaller writesets, predominance of read-only transactions). Also, we set the *maxAbortRate* parameter to the very conservative value of 1%, which should be in practice acceptable for most web applications.

In Figure 4(a) we plot the throughput (committed transactions per second) while varying the number of replicas, and the number of threads per replica. The plot shows linear speedups as the number of threads and replicas increases, highlighting the scalability of the BFC protocol. Figure 4(b) shows the performance gains achievable by BFC with respect to a classic non-voting certification scheme in terms of reduction of the execution time of write transactions, which fluctuates in the range from around 20% to around 40% and is imputable to a 3x message compression factor. This points out how the BFC scheme can achieve significant performance gains even for a negligible (i.e., 1%) additional increase of the transaction’s abort rate. This makes the BFC scheme viable, in practice, even in abort-sensitive applications.

In conclusion, the BFC scheme makes it possible to use additional replicas to improve the throughput of the system and, last but not the least, permits to use (faster) non-voting certification approaches in the presence of workloads with large readsets.

6 Conclusions and Future Work

In this paper we have overviewed some of our recent results concerning the integration of DTMs in clusters of web application servers. In particular we have reported our experiences with the development of a complex, real web application, namely *FénixEDU*, which is, to the best of our knowledge, the first web application in production to rely on (D)TM technology.

We have then focused on the description of BFC, a recently introduced certification-based transactional replication scheme that permits to reduce significantly the cost of the inter-replica synchronization phase by exploiting the space-efficient encoding of Bloom Filters.

For what concerns our ongoing and future work, we are currently pursuing several orthogonal, yet complementary, research directions, which we overview in the following.

Speculative transaction execution. The average latency of ABcast is, even for very small messages, in the order of at least a few milliseconds in typical data-center environments, see, e.g., [18, 20]. The completion time of (not replicated) TM transactions, on the other hand, is often two or three orders of magnitude smaller. Hence, in any ABcast based replication protocol, it is highly likely that transactions complete and stall (relatively) long before the ABcast is concluded.

This may lead to severe under-utilization of the available computing resources. Given the above considerations, we are currently pursuing the idea of speculatively exploring multiple alternative transaction serialization orders (rather than just the one suggested by the spontaneous order delivery as suggested in, e.g., [22]) so to maximize the utilization of any CPU core waiting for the termination of an ABcast's run.

The main challenge here is related to the fact that the number of possible serialization orders over a set composed of n elements is $n!$, which drastically reduces the probability to blindly select the correct final serialization order as the number of messages to be ordered grows. This issue raises the need for ingenious heuristics that are able to maximize the probability to drive the speculative exploration of the serialization orders towards useful trajectories.

Lease based replication mechanisms. Another orthogonal approach that we are currently pursuing is based on the idea of taking advantage from the application's data access pattern locality to reduce the frequency of activation of the ABcast-based replica synchronization schemes (and hence their inherent overhead) and to decrease the likelihood of remote conflicts.

The underlying intuition is to rely on consensus-like coordination primitives (such as the Atomic Broadcast) only to establish the ownership of a "lease" on the data items accessed by a committing transaction. On the other hand, transactions accessing data items for which the local replica already owns a lease are guaranteed not to be aborted due to a remote conflict (at least in absence of failure suspicions) and may be committed in a considerably more efficient way, avoiding the costs of ABcast-based synchronization.

By introducing the *Weak Mutual Exclusion* abstraction, see [33], we have already provided a formal specification of the lease mechanism underlying the proposed approach. At this stage, the challenge is to design and implement pragmatic, highly efficient, Weak Mutual Exclusion protocols, as well as lightweight load balancing strategies aimed at maximizing the data access pattern locality of TM applications.

Adaptive replication strategies. We hypothesize that no single universal replication scheme exists that is able to effectively cope with the high heterogeneity characterizing TM workloads. Therefore, we advocate the need for developing self-adapting TM replication schemes, able to identify in a timely and automatic way the optimal replication strategy for each incoming transaction on the basis of the (estimated) size of its readset and writeset, as well as of its conflict probability.

Implementing a polymorphic replication strategy requires facing two main challenges: on one hand, ensuring the consistent interaction of different replication algorithms and, on the other hand, engineering lightweight and timely mechanisms to identify automatically the characteristics of incoming transactions [15] and the corresponding preferable replication scheme.

References

1. Adl-Tabatabai, A.-R., Kozyrakis, C., Saha, B.: Unlocking concurrency. *ACM Queue* 4(10), 24–33 (2007)
2. Amza, C., Cox, A.L., Zwaenepoel, W.: Data replication strategies for fault tolerance and availability on commodity clusters. In: *Proc. of the Conference on Dependable Systems and Networks (DSN)*, pp. 459–472 (2000)
3. Bernstein, P.A., Hadzilacos, V., Goodman, N.: *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, Reading (1987)
4. Bocchino, R.L., Adve, V.S., Chamberlain, B.L.: Software transactional memory for large scale clusters. In: *Proc. of the Symposium on Principles and Practice of Parallel Programming (PPOPP)*, pp. 247–258. ACM, New York (2008)
5. Broder, A., Mitzenmacher, M.: Network Applications of Bloom Filters: A Survey. *Internet Mathematics* 1(4), 485–509 (2003)
6. Cachopo, J., Rito-Silva, A.: Combining software transactional memory with a domain modeling language to simplify web application development. In: *Proc. of the International Conference on Web Engineering (ICWE)*, pp. 297–304 (2006)
7. Cachopo, J., Rito-Silva, A.: Versioned boxes as the basis for memory transactions. *Sci. Comput. Program.* 63(2), 172–185 (2006)
8. Cardellini, V., Casalicchio, E., Colajanni, M., Yu, P.S.: The state of the art in locally distributed web-server systems. *ACM Comput. Surv.* 34(2), 263–311 (2002)
9. Carvalho, N., Cachopo, J., Rodrigues, L., Rito Silva, A.: Versioned Transactional Shared Memory for the FenixEDU Web Application. In: *Proc. of the Workshop on Dependable Distributed Data Management (WDDDM)*. ACM, New York (2008)
10. Carvalho, N., Pereira, J., Rodrigues, L.: Towards a generic group communication service. In: *Proc. of the International Symposium on Distributed Objects and Applications, DOA* (2006)
11. Cattell, R.G.G., Barry, D.K., Berler, M., Eastman, J., Jordan, D., Russell, C., Shadow, O., Stanienda, T., Velez, F. (eds.): *The Object Data Standard – ODMG 3.0*. Morgan Kaufmann Publishers, Inc, Los Altos (2000)
12. Couceiro, M., Romano, P., Carvalho, N., Rodrigues, L.: D²STM: Dependable Distributed Software Transactional Memory. In: *Proc. of the Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE Computer Society Press, Los Alamitos (2009)
13. Defago, X., Schiper, A., Urban, P.: Total order broadcast and multicast algorithms: Taxonomy and survey. *ACM Computing Surveys* 36(4), 372–421 (2004)
14. Frølund, S., Guerraoui, R.: e-Transactions: End-to-end reliability for three-tier architectures. *IEEE Transaction on Software Engineering* 28(4), 378–395 (2002)
15. Garbatov, S., Cachopo, J., Pereira, J.: Data access pattern analysis based on bayesian updating. In: *Proc. of the 1st Simpósio de Informática (INForum)*, Lisbon, Portugal (September 2009)
16. Gray, J., Helland, P., O’Neil, P., Shasha, D.: The dangers of replication and a solution. In: *Proc. of the Conference on the Management of Data (SIGMOD)*, pp. 173–182. ACM, New York (1996)
17. Guerraoui, R., Kapalka, M., Vitek, J.: STMBench7: a benchmark for software transactional memory. *SIGOPS Oper. Syst. Rev.* 41(3), 315–324 (2007)
18. Guerraoui, R., Levy, R.R., Pochon, B., Quema, V.: High throughput total order broadcast for cluster environments. In: *Proc. of the International Conference on Dependable Systems and Networks*, pp. 549–557. IEEE Computer Society, Los Alamitos (2006)
19. Guerraoui, R., Rodrigues, L.: *Introduction to Reliable Distributed Programming*. Springer, Heidelberg (2006)

20. Kaashoek, M., Tanenbaum, A.: An evaluation of the Amoeba group communication system. In: Proc. of the International Conference on Distributed Computing Systems (ICDCS), p. 436. IEEE Computer Society, Los Alamitos (1996)
21. Keleher, P., Cox, A.L., Zwaenepoel, W.: Lazy release consistency for software distributed shared memory. In: Proc. of the International Symposium on Computer Architecture (ISCA), pp. 13–21. ACM, New York (1992)
22. Kemme, B., Pedone, F., Alonso, G., Schiper, A.: Processing transactions over optimistic atomic broadcast protocols. In: Proc. of the International Conference on Distributed Computing Systems (ICDCS), p. 424. IEEE Computer Society, Los Alamitos (1999)
23. Kotselidis, C., Ansari, M., Jarvis, K., Lujan, M., Kirkham, C., Watson, I.: DiSTM: A software transactional memory framework for clusters. In: Proc. of the International Conference on Parallel Processing (ICPP), pp. 51–58 (2008)
24. Li, K., Hudak, P.: Memory coherence in shared virtual memory systems. In: Proc. of the Symposium on Principles of Distributed Computing (PODC), pp. 229–239. ACM, New York (1986)
25. Miranda, H., Pinto, A., Rodrigues, L.: Appia, a flexible protocol kernel supporting multiple coordinated channels. In: Proc. International Conference on Distributed Computing Systems (ICDCS), pp. 707–710. IEEE, Los Alamitos (2001)
26. OJB. Object/Relational Bridge - OJB (2007), <http://db.apache.org/ojb>
27. Patino-Martínez, M., Jiménez-Peris, R., Kemme, B., Alonso, G.: Scalable replication in database clusters. In: Herlihy, M.P. (ed.) DISC 2000. LNCS, vol. 1914, pp. 315–329. Springer, Heidelberg (2000)
28. Pedone, F., Guerraoui, R., Schiper, A.: The database state machine approach. *Distributed and Parallel Databases* 14(1), 71–98 (2003)
29. Perez-Sorrosal, F., Patino-Martinez, M., Jimenez-Peris, R., Kemme, B.: Consistent and scalable cache replication for multi-tier J2EE applications. In: Cerqueira, R., Campbell, R.H. (eds.) *Middleware 2007*. LNCS, vol. 4834, pp. 328–347. Springer, Heidelberg (2007)
30. Rodrigues, L., Miranda, H., Almeida, R., Martins, J., Vicente, P.: The GlobData fault-tolerant replicated distributed object database. In: Shafazand, H., Tjoa, A.M. (eds.) *EurAsia-ICT 2002*. LNCS, vol. 2510, pp. 426–433. Springer, Heidelberg (2002)
31. Romano, P., Carvalho, N., Rodrigues, L.: Towards distributed software transactional memory systems. In: Proc. of the Workshop on Large-Scale Distributed Systems and Middleware, LADIS (2008)
32. Romano, P., Quaglia, F., Ciciani, B.: A lightweight and scalable e-Transaction protocol for three-tier systems with centralized back-end database. *IEEE Transactions on Knowledge and Data Engineering* 17(11), 1578–1583 (2005)
33. Romano, P., Rodrigues, L., Carvalho, N.: The weak mutual exclusion problem. In: Proc. 23rd IEEE International Parallel and Distributed Processing Symposium. IEEE Computer Society Press, Los Alamitos (to appear)
34. Stonebraker, M., Madden, S., Abadi, D.J., Harizopoulos, S., Hachem, N., Helland, P.: The end of an architectural era: (it's time for a complete rewrite). In: Proc. of the 33rd international conference on Very Large Data Bases (VLDB), pp. 1150–1160. VLDB Endowment (2007)
35. Terracotta Inc. Terracotta, <http://www.terracotta.org/>
36. Transaction Processing Performance Council. TPC Benchmark™ W, Standard Specification, Version 1.8. Transaction Processing Performance Council (2002)
37. Transaction Processing Performance Council. TPC Benchmark™ TPC-APP, Standard Specification, Version 1.0. Transaction Processing Performance Council (2004)

Optimizing Distributed Execution of WS-BPEL Processes in Heterogeneous Computing Environments

Qishi Wu¹, Yi Gu¹, Liang Bao², Wei Jia², Huichen Dai², and Ping Chen²

¹ Dept of Computer Science, University of Memphis, Memphis, TN 38016, USA
{qishiwu,yigu}@memphis.edu

² College of Software, XiDian University, Xi'an, Shanxi, 710071, China
{baoliang,weijia,huichendai,pingchen}@mail.xidian.edu.cn

Abstract. Workflow-structured Web service composition is an emerging computing paradigm for constructing next-generation large-scale distributed applications within and across organizational boundaries. Mapping such application workflows in heterogeneous environments and optimizing their performance in terms of quick response and high scalability are vital to the success of these distributed applications. Workflows with complex execution semantics and dependencies are typically modeled as directed acyclic graphs. We construct cost models to estimate data processing and transfer overheads and formulate the restricted workflow mapping for minimum total execution time as an NP-complete optimization problem. We propose a heuristic approach to this problem that recursively computes and maps the critical path to network nodes using a dynamic programming-based procedure. The performance superiority of the proposed approach is illustrated by an extensive set of simulations and further verified by experimental results from a real network in comparison with existing methods.

Keywords: WS-BPEL, workflow mapping, optimization, heuristic algorithm.

1 Introduction

As the number of Web services of wide variety grows rapidly in the Internet, Web service composition has become an important computing paradigm for constructing next-generation large-scale distributed applications within and across organizational boundaries. Successful business operations require an efficient and flexible scheme for pooling globally available Web service-based resources together to quickly adapt to various customer needs and dynamic market conditions. WS-BPEL (Web Service Business Process Execution Language) is now a de facto specification for Web service composition.

A WS-BPEL application based on composite Web services features complex execution semantics and is typically coordinated by a single node referred to as a centralized orchestrator. The WS-BPEL process is usually designed by application developers according to certain business logics and manually deployed on a WS-BPEL engine. Fig. 1 diagrammatizes a typical execution setup of WS-BPEL process: a request is sent to the centralized WS-BPEL engine, which orchestrates the invocation of Web services located in different Web containers. As pointed out in [7], instead of transferring data directly from the point of generation to the point of consumption, this execution model

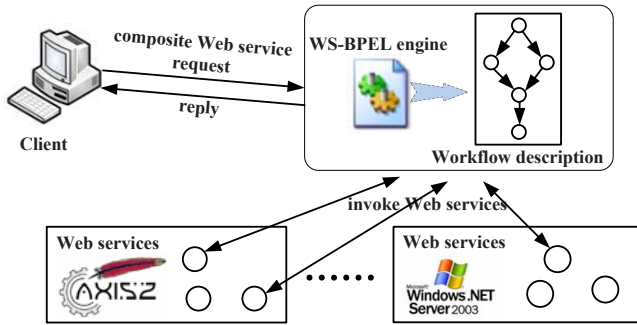


Fig. 1. A typical execution setup of WS-BPEL process

uses the engine as a central intermediary to exchange data between Web services, resulting in unnecessary network traffic. In addition, a Web service could generate a large volume of data that are irrelevant to the composite service but still need to be transferred to the engine where they are eventually discarded, hence causing unnecessary workload in the network. This client-server communication model poses an inherent performance limitation on scalability: the system performance degrades significantly as the traffic and workload increase in heterogeneous and cross-organization environments.

We propose a distributed approach to execute a WS-BPEL process that overcomes the above performance limitation by constructing and mapping a WS-BPEL workflow with direct inter-web service data transfer to a set of network nodes. The workflow is constructed by a static analyzer that takes three steps: (i) load a WS-BPEL process and transform it into a customized memory structure of WS-BPEL process (Java classes); (ii) load the generated memory representation of BPEL process and transform it into a BCFG (BPEL Control Flow Graph) representation; and (iii) apply a revised algorithm in [3] to remove control flow edges in BCFG and insert control and data dependence edges, which generates the final Program Dependence Graph (PDG) representation of the WS-BPEL process. We apply static analysis techniques of multi-threaded programs to BPEL process and some composition structure patterns, such as AND split (fork), XOR split (conditional), loop and AND join (merge), and XOR join (trigger), which can be modeled and represented in our tCFG (threaded Control Flow Graph)-structured BCFG [3, 10]. Note that loop operations can be managed by unfolding the cycles as proposed in [31]. We would like to point out that the activities in BPEL considered here are synchronous and stateless Web services invocations. Invocations of asynchronous and stateful Web services are more complicated and are out of the scope of this paper.

The workflow mapping may be subject to some restrictions and involves two types of graphs: (i) a Directed Acyclic Graph (DAG) that models the workflow of a WS-BPEL process, where each vertex represents an activity and each directed edge represents the data transfer or execution dependency between two activities (also referred to as PDG [12]); (ii) a directed weighted graph that represents an underlying physical computer network, where Web services are deployed on heterogeneous computing nodes that are connected by network links with different bandwidths. The topology of the computer network may not be complete in a dedicated network environment or even

in the Internet due to different administrative policies and firewall settings. Furthermore, the Web services in the Internet come and go dynamically while those deployed in high-speed reliable enterprise intranet are more stable and predicable. Mapping such workflows into heterogeneous computing environments and optimizing their end-to-end performance are crucial to ensuring the success of business processes requiring quick response and the maximum utilization of system resources.

The workflow-structured WS-BPEL process requires distributed execution of complex Web service components with inter-component communications using massively dispersed computing and networking resources to support business collaborations in various domains. The workflow mapping objective is to strategically select an appropriate set of network nodes that host different Web services in the physical computer network and assign each activity in the WS-BPEL process to one of those selected nodes to achieve the Minimum Execution Time (MET) of the process for fast response. Certain activities in a WS-BPEL process might be restricted to some specific computing nodes providing the corresponding Web services. We refer to such activities as restricted activities as opposed to free activities, which can be mapped onto any computing nodes. We allow multiple activities to be mapped onto the same node and the computing resources of that node are shared in a fair manner by those activities running concurrently on that node. Note that activities assigned to the same node do not share computing resources if their executions do not overlap due to the dependency or unavailability of input data. Similarly, the bandwidth of a network link is fairly shared by multiple data transfers that take place concurrently on the same link. We formulate the workflow mapping with arbitrary node reuse and certain mapping restrictions as an NP-complete optimization problem, and propose a heuristic approach, *restricted Recursive Critical Path* (rRCP), which is modified from the Recursive Critical Path (RCP) algorithm in [29] by taking the mapping restrictions into consideration.

The rest of the paper is organized as follows. We conduct an extensive survey of WS-BPEL processes and workflow mapping in Section 2. We construct mathematical models and formulate the problem in Section 3. In Section 4, we design the rRCP algorithm for workflow mapping to achieve MET. The implementation details and performance evaluations are presented in Section 5. We conclude our work in Section 6.

2 Related Work

Web services have found pervasive applications in different domains over wide-area networks [17, 22, 27]. Guo *et al.* proposed the ANGEL model for service composition and adopted a redundant mechanism in ANGEL to improve system availability [17]. In [27], Shin *et al.* proposed a simple heuristic solution to Web service composition where the highest search priority is given to services providing the largest number of new responses. Li *et al.* proposed a general purpose Web Service Management System in [22] that enables execution optimization of composite services through multiple engines. In the Symphony project [24], Mangla [25] partitioned a composite Web service written as a single WS-BPEL program into an equivalent set of decentralized processes using a new code partitioning algorithm based on PGD to minimize communication costs and maximize the throughput of multiple concurrent instances of the input program. However, Mangla's work does not consider the situation where multiple services

may be executed on a single server. Yildiz *et al.* proposed an efficient process transformation technique that converts a process conceived for centralized execution to a set of nested processes to be deployed on dynamically bound services [30]. Other research efforts along this line include the static optimization of WS-BPEL process [3] and batch invocation of Web services [10], where the former applies static analysis to the WS-BPEL process to identify “concurrent branches” and the latter reduces the number of connections by forming batch invocation request to implement “one request, many invocations of Web services”. Security and performance issues of BPEL processes were studied in [5] and [6], respectively.

The workflow mapping problem in distributed network environments under different constraints has been extensively studied by researchers in various disciplines [4, 8, 9, 15, 28] and continues to be the focus of distributed computing due to its theoretical significance and practical importance. Zhu *et al.* proposed a model of overlay network with linear capacity constraints (LCC) [32], which incorporates correlated link capacities by formulating shared bottlenecks as linear constraints of link capacities. Guerin *et al.* tackled an all hops optimal path problem to minimize end-to-end delay or maximize bandwidth with a limit on the maximum number of possible hops [16]. Among the traditional workflow mapping problems in theoretical aspects of computing, subgraph isomorphism is known to be NP-complete [14] while the complexity of graph isomorphism still remains open. Many special cases of graph isomorphism under different topology constraints on the mapped (workflow) or mapping (network) graphs can be solved in polynomial time, including isomorphism between planar graphs [18] and bounded valence graphs [23]. The mapping computational complexity could also be reduced by introducing an adequate representation of the search space and process, and pruning unprofitable search paths in the search space [13].

Many research efforts have been focused on static scheduling algorithms for multiprocessors that are considered as identical resources. Kwok *et al.* proposed Dynamic Critical-Path (DCP) scheduling algorithm [20] to map task graphs with arbitrary computation and communication costs to a multiprocessor system with an unlimited number of identical processors in a fully-connected network. A task graph scheduling scheme for streaming data, *Streamline*, which places a coarse-grain dataflow graph on available grid resources, is proposed in [2] to improve the performance of graph mapping for streaming applications with various demands in distributed network environments. Most graph mapping or task scheduling problems in grid environments assume complete networks with heterogeneous resources. Similar mapping problems are also studied in the context of sensor networks. Sekhar *et al.* proposed an optimal algorithm for mapping subtasks onto a large number of sensor nodes based on an A^* algorithm, and also proposed a greedy A^* algorithm to reduce the complexity of the original optimal solution accounting for the limited energy of each sensor node [26].

3 Cost Model and Problem Formulation

We model the workflow of a WS-BPEL process as a task graph $G_t = (V_t, E_t)$, $|V_t| = m$, where vertices represent different computing activities: w_0, w_1, \dots, w_{m-1} . The data or control dependency between a pair of adjacent vertices w_i and w_j is represented by a

directed edge $e_{i,j}$ with data size $z_{i,j}$ between them and the entire workflow is modeled as a DAG starting from the source activity w_0 and ending at the destination activity w_{m-1} . An intermediate activity w_i cannot start any processing until it receives all required input data from its preceding activities. The computational complexity of an activity is modeled as a function $f_{w_i}(\cdot)$ on the total aggregated input data z_{w_i} , and the activity sends results to its succeeding activities once it completes the required processing. We estimate the computing time of an activity w_i running on network node v_j as $T_{comp}(w_i, v_j) = \frac{f_{w_i}(z_{w_i})}{p_j}$. The actual runtime of an activity does not only depend on the total aggregated incoming data size and computational complexity, but also the capacity of system resources deployed on the selected nodes as well as their availability during the runtime. Note that for an application with multiple source or destination activities, we could convert it to this model by inserting a virtual starting or ending activity of complexity zero connected to all source or destination activities with zero-sized output or input data transfers.

Table 1. Workflow and network parameters

Parameters	Definitions
$G_t = (V_t, E_t)$	task graph
m	number of activities in the workflow
w_i	the i -th computing activity
$e_{i,j}$	dependency edge from activity w_i to w_j
$z_{i,j}$	data size of dependency edge $e_{i,j}$
z_{w_i}	aggregated input data size of activity w_i
$f_{w_i}(\cdot)$	computational complexity of activity w_i
$G_c = (V_c, E_c)$	computer network graph
n	number of nodes in the network graph
v_i	the i -th network or computer node
v_s	source node
v_d	destination node
p_i	normalized computing power of v_i
$l_{i,j}$	network link between nodes v_i and v_j
$b_{i,j}$	bandwidth of link $l_{i,j}$
$d_{i,j}$	minimum link delay of link $l_{i,j}$
$T_{comp}(w_i, v_j)$	computing time of activity w_i running on node v_j
$T_{trans}(z_{h,k}, l_{i,j})$	transfer time of data $z_{h,k}$ over link $l_{i,j}$
T_{total}	total execution time required for a WS-BPEL process

The underlying computer network is modeled as an arbitrary weighted graph $G_c = (V_c, E_c)$ consisting of $|V_c| = n$ computer nodes interconnected by directed communication links represented by a matrix $L[n \times n]$. The processing power of a computer node is a complex notion that combines a variety of host factors such as processor frequency, bus speed, memory size, I/O performance, and presence of co-processors. For simplicity, we use a normalized variable p_i to represent the overall processing power of a network node v_i without specifying its detailed system resources. There are two

parameters, bandwidth (BW) $b_{i,j}$ and minimum link delay (MLD) $d_{i,j}$, associated with a network link $l_{i,j} \in L$ ($i, j \in n$) between nodes v_i and v_j . The estimated time of transferring the data $z_{h,k}$ between modules w_h and w_k over the network link $l_{i,j}$ can be calculated as $T_{trans}(z_{h,k}, l_{i,j}) = \frac{z_{h,k}}{b_{i,j}} + d_{i,j}$.

For convenience, we tabulate in Table 1 the notations we define in the above workflow and network models to facilitate the problem formulation. A mapping example is illustrated in Fig. 2 where activities w_h and w_g are mapped to network node v_i , w_l is mapped to v_j , and w_r , w_t and w_u are mapped to v_k . The dashed arrows represent the data or control dependencies in a WS-BPEL process and the solid arrows represent the communication links between network nodes. Activity w_r cannot start its execution until it receives all required data from its preceding activities w_h and w_l . Note that w_r does not receive data directly from w_h and w_l in the centralized execution model, where a central engine is responsible for all data communication. Activity w_r aggregates incoming data and performs a predefined computing routine whose complexity is modeled as function $f_{w_r}(\cdot)$ on the total aggregated input data z_{w_r} and sends out the results to its succeeding activities upon finishing its processing.

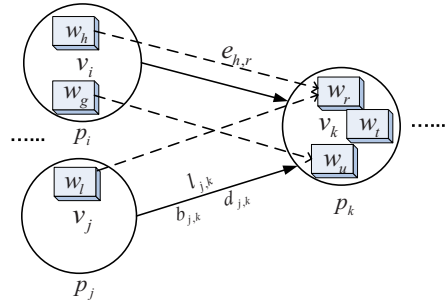


Fig. 2. A mapping example

The mapping objective is to map all the activities of a WS-BPEL process onto an appropriate set of computer nodes to minimize total execution time T_{total} , which is determined by its critical path (CP), i.e. the longest path of the workflow. Once a mapping scheme is determined, one may calculate T_{total} by adding up all the computing time and transfer time incurred on the CP, which can be estimated as:

$$\begin{aligned}
 T_{Total} &= \sum_{w_i \in CP} T_{comp}(w_i, v_h) + \sum_{e_{j,k} \in CP} T_{trans}(z_{j,k}, l_{f,g}) \\
 &= \sum_{w_i \in CP} \frac{f_{w_i}(z_{w_i})}{p_h} + \sum_{e_{j,k} \in CP} \left(\frac{z_{j,k}}{b_{f,g}} + d_{f,g} \right)
 \end{aligned}
 \tag{1}$$

We assume that the inter-activity transfer time on the same node is negligible considering that the in-memory transfer rate is much faster than across networks.

The proposed workflow mapping problem considers node reuse and resource share. In the underlying network, multiple services might be mapped onto the same node but some services are only available on certain nodes. To simplify the time estimation of an activity, we combine the time cost for service invocation and activity processing.

4 Restricted Workflow Mapping Algorithm

The workflow mapping or scheduling problem is known to be NP-complete [2, 21] even on two processors without any topology or connectivity restrictions [1]. The mapping problem in this paper considers mapping restrictions: some activities in the WS-BPEL

process can only be mapped onto certain nodes with specific resources to support the execution of those restricted activities. We modify and adapt the Recursive Critical Path (RCP) algorithm in [29] to this new problem and propose a restricted version of RCP algorithm, referred to as *restricted Recursive Critical Path* (rRCP).

4.1 rRCP Algorithm

rRCP features a recursive optimization strategy. In each round, it chooses the CP based on the previous round of calculation as shown in Fig. 3 and maps it to the network nodes using a dynamic programming-based procedure until the mapping results converge to an optimal or suboptimal point or a certain termination condition is met. The mapping restrictions are taken into consideration when each activity is being mapped.

The pseudocode of the rRCP mapping scheme is provided in Alg. 1. The initial mapping assumes resource homogeneity and connectivity completeness in computer network, that is, the computer network is considered as complete with identical computer nodes and communication links. Thus, we only need to consider the workflow when calculating the initial computing and transfer time cost components on each activity and over each dependency edge, respectively. With the initial time cost components in workflow G_t^1 , we find its CP P_1 using a procedure defined in $FindCriticalPath()$, which essentially finds the longest path in a DAG. From this point on, we remove the assumption on resource homogeneity and connectivity completeness, and map the current CP, i.e P_1 to the real computer network using a dynamic programming-based pipeline mapping algorithm $MapCriticalPath()$ with arbitrary node reuse as well as mapping restrictions for MET. The activities that are not located on the CP, referred to as branch or non-critical activities, are mapped to the network nodes using a procedure defined in $MapNonCriticalActivity()$. Based on the current mapping, we compute a new CP using updated time cost components in G_t^i and calculate a new MET. The above steps are repeated until a certain condition is met, for example, the difference between two METs of two consecutive iterations is less than a preset threshold.

The complexity of the rRCP algorithm is $O(k(m + |E_t|) \cdot |E_t|)$, where m represents the number of activities in the WS-BPEL process, $|E_t|$ and $|E_c|$ denote the number of dependency edges in the workflow and communication links in the computer network, respectively, and k is the number of iterations where CPs are calculated and mapped.

The algorithm for CP calculation is well studied and documented in the literature. The algorithms for CP mapping $MapCriticalPath()$ and non-critical activities mapping $MapNonCriticalActivity()$ are similar to those proposed in [29] using a dynamic programming-based and a greedy-based procedure, respectively. Note that when

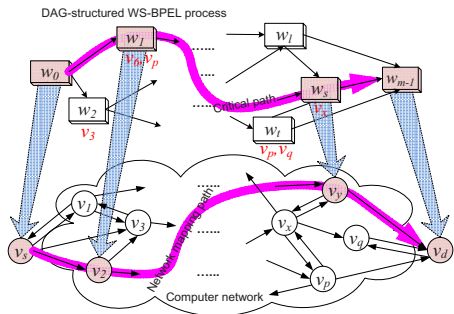


Fig. 3. An example of WS-BPEL process mapping using rRCP algorithm

Algorithm 1. $\text{rRCP}(G_t, G_c, v_s, v_d)$

```

1:  $MET_0 = MET_{max} = MaxValue$ ;
2: Create  $G_c^*$  by assuming resource homogeneity and connectivity completeness in  $G_c$ ;
3: Calculate initial cost components for  $G_t^1$  based on  $G_c^*$ ;
4:  $P_1 = FindCriticalPath(G_t^1, w_0, w_{m-1})$ ;
5:  $MET_1(G_t^1) = \sum(T_{comp}(P_1) + T_{trans}(P_1))$ ;
6:  $i = 1$ ;
7: while  $|MET_i - MET_{i-1}| \geq Threshold$  do
8:   Call  $MapCriticalPath(P_i, G_c, v_s, v_d)$  to map the activities on CP  $P_i$  to network  $G_c$  with mapping restrictions;
9:   Call  $MapNonCriticalActivity(P_i, G_t^i, G_c, v_s, v_d)$  to map the activities not on CP to network  $G_n$  with mapping restrictions;
10:   $i = i + 1$ ;
11:  Calculate new time cost for  $G_t^i$  based on the current mapping;
12:   $P_i = FindCriticalPath(G_t^i, w_0, w_{m-1})$ ;
13:   $MET_i(G_t^i) = \sum(T_{comp}(P_i) + T_{trans}(P_i))$ ;
14: return  $MET_i(G_t^i)$ .

```

multiple activities are assigned to the same computer node, resources on this node are shared among these activities only if they can run concurrently. Two activities are considered “independent” if there does not exist any path between them, and only independent activities may run concurrently on the same node. It is worth pointing out that the time calculation based on this resource share strategy is still an approximation since the execution start time of an activity depends on the arrival time of its latest input data. Therefore, even independent activities deployed on the same node may not run concurrently if their execution start and end times do not overlap. Note that some activities in the WS-BPEL process can only be executed on a subset of computers in the network, which imposes additional constraints for selecting nodes. In Fig. 3, the IDs listed under an activity are the IDs of those computer nodes that have been ruled out for deploying that activity. For example, activity w_1 cannot be mapped to nodes v_6 and v_p .

5 Performance Evaluation

Despite the widespread application of WS-BPEL processes in a wide spectrum of fields, there still lacks a standardized benchmark for evaluating their performances. We present below the results from both simulations and real network experiments to illustrate the performance superiority of the proposed mapping solution over existing algorithms.

5.1 Simulation Results

The proposed rRCP algorithm is implemented in C++ and runs on a Windows XP desktop PC equipped with a 3.0 GHz CPU and 2 Gbytes memory. For performance comparison purposes, we also implement the other three algorithms, namely, Greedy A^* , Streamline, and Naive Greedy. A^* algorithm is a static allocation scheme proposed by

Sekhar *et al.* [26], which maps the subtasks of a DAG-like workflow onto a large number of sensor nodes. A greedy A^* algorithm, which is specifically designed to reduce the complexity of the A^* algorithm, explores only the least-cost path of the search tree in

Table 2. Simulation-based performance comparison of MET among four algorithms

Prb Idx	Problem Size $m, E_t , n, E_c $	MET (s)			
		rRCP	Greedy A^*	Streamline	Naive Greedy
1	4, 6, 6, 35	1.05	1.08	1.15	1.08
2	6, 10, 10, 96	1.15	1.85	1.33	1.23
3	10, 18, 15, 222	1.59	1.89	1.95	1.92
4	13, 24, 20, 396	1.49	2.16	2.09	2.19
5	15, 30, 25, 622	2.29	2.57	2.67	2.32
6	19, 36, 28, 781	1.41	1.75	1.71	1.57
7	22, 44, 31, 958	1.17	1.43	1.61	1.74
8	26, 50, 35, 1215	3.14	3.76	3.83	3.57
9	30, 62, 40, 1598	4.40	5.38	5.41	4.92
10	35, 70, 45, 2008	4.24	5.19	5.99	4.48
11	38, 73, 47, 2200	3.21	3.64	5.16	4.40
12	40, 78, 50, 2478	2.69	3.73	4.31	3.17
13	45, 96, 60, 3580	1.41	1.52	2.07	1.81
14	50, 102, 65, 4220	1.99	5.01	3.87	4.59
15	55, 124, 70, 4890	7.64	12.35	9.49	10.84
16	60, 240, 75, 5615	9.98	11.45	15.07	13.55
17	75, 369, 90, 8080	11.57	19.37	14.68	15.13
18	80, 420, 100, 9996	24.83	31.73	30.69	28.50
19	90, 500, 150, 22496	17.33	24.74	20.77	21.37
20	100, 660, 200, 39990	35.79	41.37	38.66	39.29

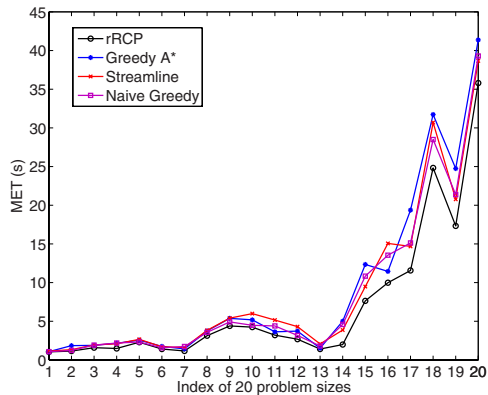


Fig. 4. Simulation-based MET performance comparison among the four algorithms

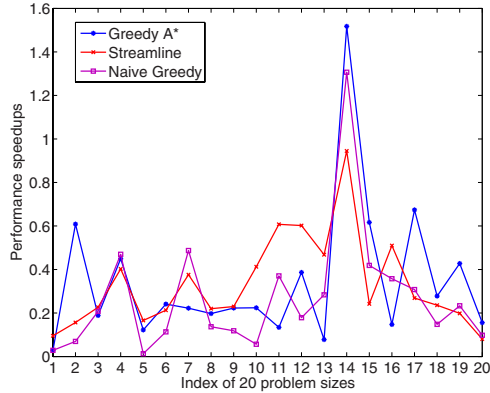


Fig. 5. Performance speedups of rRCP over the other three algorithms

Table 3. Simulation-based MET performance comparison of mean and standard deviation

Prb Idx	Problem Size $m, E_t , n, E_c $	MET (s)							
		rRCP		Greedy A*		Streamline		Naive Greedy	
		Mean	Std Div	Mean	Std Div	Mean	Std Div	Mean	Std Div
1	4, 6, 6, 35	0.5280	0.3824	0.5600	0.4121	0.5600	0.4105	0.5600	0.4121
2	10, 18, 15, 222	1.2750	0.4832	1.6120	0.4893	1.4680	0.7200	1.5530	0.6351
3	15, 30, 25, 622	2.0600	0.4323	2.0090	0.4967	2.4430	0.5029	2.3080	0.7288
4	22, 44, 31, 958	2.1160	0.5808	2.7720	0.8537	2.9720	0.8707	2.3200	0.7722
5	30, 62, 40, 1598	2.9780	1.4326	3.7160	1.8126	4.2450	1.4535	3.1270	1.4951
6	40, 78, 50, 2478	3.0360	1.2833	3.8980	1.4230	4.7650	1.5064	3.4040	1.8909
7	50, 102, 65, 4220	3.6840	1.1972	4.6110	1.6615	5.2930	1.3297	3.8940	1.2713
8	60, 240, 75, 5615	8.8360	2.0971	11.9580	3.0178	12.3640	2.2823	10.0900	2.3685
9	80, 420, 100, 9996	16.1200	2.5483	21.3010	2.9096	21.4230	3.6038	20.1380	5.1338
10	100, 660, 200, 39990	25.1450	4.4816	30.0550	6.6525	28.8300	5.1543	26.5470	6.5483

the solution space, instead of searching all feasible paths, assuming that the optimal solution is most likely to be found on this path. Streamline works as a global greedy algorithm that expects to maximize the throughput of a distributed application by assigning the best resources to the most needy stages in terms of computation and communication requirements at each step [2]. The greedy algorithm makes an activity mapping decision at each step only based on the current information without considering the effect of this local decision on the mapping performance at later steps.

We conduct an extensive set of mapping experiments for MET using a large number of simulated workflows for WS-BPEL processes and computer networks. These simulation datasets are generated by randomly varying the parameters of the workflows and computer networks within a suitably selected range of values: (i) the number of activities and the complexity of each activity, (ii) the number of inter-activity communications and the data or control flow between two activities, (iii) the number of nodes and the

processing power of each node, and (iv) the number of links and the BW and MLD of each link. The topology and size of 20 simulated computing workflows and computer networks as well as the MET calculated by four mapping algorithms in comparison are tabulated in Table 2, where the problem size is represented by a four-tuple: m activities and $|E_t|$ edges in the workflow, and $|n|$ nodes and $|E_c|$ links in the computer network. For a visual performance comparison, we plot in Fig. 4 the MET performance measurements from these four algorithms for 20 different problem sizes ranging from small to large scales. We observe that rRCP exhibits comparable or superior MET performances over the other three algorithms. Note that the MET measurement points plotted along the x axis (index of problem size) are independent of each other due to the random generation of these 20 problem instances. However, since MET represents the total execution latency from source to destination, a larger problem size with more network nodes and computing activities generally, not absolutely though, incurs a longer mapping path resulting in a longer execution time, as the overall increasing trend indicates.

We also plot the MET performance speedup of rRCP over the other three algorithms in Fig. 5 which is defined as: $Speedup = \left| \frac{MET_{rRCP} - MET_{other}}{MET_{rRCP}} \right|$, where MET_{rRCP} represents the MET for rRCP and MET_{other} denotes MET for each of the other three algorithms in comparison. We observe that rRCP achieves an average performance improvement around 20%-80% in most of the cases and even more than 150% speedups in some cases, which demonstrates the MET performance superiority of the rRCP algorithm.

To further investigate the robustness of these mapping algorithms, for each of 10 problem sizes chosen from the previous 20 cases, we randomly generate 20 problem instances and run four mapping algorithms on them. We then calculate and plot the mean value and standard deviation over 20 instances for each problem size in Table 3 and Fig. 6. We observe that rRCP achieves the best MET performance in an expected sense with the smallest standard deviation, which demonstrates the performance robustness and optimization stability of rRCP in achieving MET in various workflows and computer networks of disparate topologies and different scales.

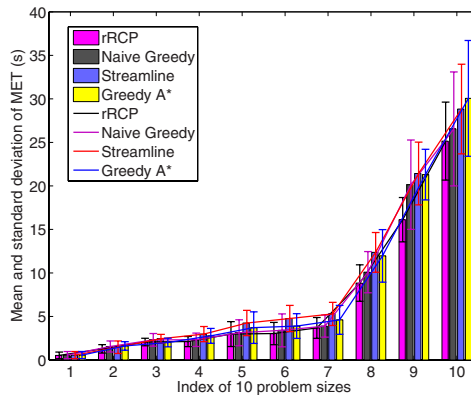


Fig. 6. Mean and standard deviation of MET performance of four algorithms

5.2 Experimental Results

We also conduct experiments on workflow deployment and WS-BPEL process execution in real networks. The experimental settings involve 10 Intel-based Windows machines labeled from 0 to 9, each of which runs ActiveBPEL, an open source WS-BPEL engine [11]. The hardware and software configurations of each computer are provided in Table 4. These computers are connected via a reliable and fast local-area network.

We execute two groups of processes in this experimental network environment: (i) The first group consists of four example services defined in OASIS WS-BPEL 2.0 Standard [19] with slight modification, i.e. Shipping Service, Ordering Service, Loan Approval Service, and Auction Service. These processes involve a relatively small number of activities. (ii) The second group consists of six typical WS-BPEL processes, each of which falls in one of

these categories with unique characteristics: computation-intensive, service-invocation-intensive, and the combination of them. For example, the Office Automation and Draining System processes, the Tool Integration and Travel Reserve, and the Online Book Purchase and Train Tickets belong to the first, second and third category, respectively.

We deploy and execute the activities of each process to a computer according to the mapping scheme computed by one of four mapping algorithms, and measure the corresponding MET as shown in Table 5. We observe that rRCP algorithm outperforms the other three algorithms in terms of real MET measurements, which is consistent with the simulation results. Qualitatively similar results are obtained from larger-scale processes in the second group. The experimental results based on these two groups of processes illustrate the performance superiority of rRCP algorithm in real network environments. Due to the limit on available physical resources, the problems of large scales as in the simulations are not tested.

We also investigate the performance comparison between distributed BPEL processes using the rRCP mapping scheme and traditional centralized execution (CntrExe) processes. The MET measurements in real networks and their corresponding simulation results are provided in Table 6. We observe that BPEL processes using the rRCP

Table 4. Specifications of 10 computers used in the experiments

No.	CPU (GHz)	RAM (GB)	OS
0-5	2.5 x 2	1.99	Windows XP
6	1.8 x 2	0.99	Windows XP
7	2.8	1	Windows XP
8	2.8	1	Windows XP
9	2.8	1.5	Windows XP

Table 5. Experiment-based MET performance comparison of BPEL processes

Prb Idx	Problem Size $m, E_t , n, E_c $	MET (s)			
		rRCP	Greedy A^*	Streamline	Naive Greedy
1	3, 2, 10, 98	70.03	70.03	83.52	70.03
2	5, 4, 10, 98	72.26	76.58	107.19	78.11
3	5, 5, 10, 98	105.76	113.16	134.52	113.14
4	14, 16, 10, 98	199.06	294.23	361.22	263.82

Table 6. MET comparison between distributed BPEL processes and centralized execution using both experiments and simulations

Prb Idx	Problem Size $m, E_t , n, E_c $	MET (s) using rRCP		
		BPEL Process (experiments)	CntrExe Process (experiments)	BPEL Process (simulations)
1	3, 2, 10, 98	70.03	106	31.95
2	5, 4, 10, 98	72.26	215	32.71
3	5, 5, 10, 98	105.76	292	71.41
4	14, 16, 10, 98	199.06	355	102.04

mapping scheme achieve 2-3 times MET performance improvements over centralized execution processes. The real MET measurements are generally larger than the simulation results since the latter does not consider network overheads, system dynamics, and the CP is an approximation of MET.

6 Conclusion

We tackled the problem of mapping the workflow of a BPEL process to the computer network to achieve MET and formulated it as a restricted workflow mapping optimization problem. We constructed mathematical models for BPEL processes and computer networks, and proposed rRCP algorithm with mapping restrictions of certain activities. The performance superiority of the rRCP algorithm was justified by both extensive simulation and experimental results. The activities considered in this paper are only synchronous and stateless Web services invocations. We will investigate the invocations of asynchronous and stateful Web services for performance improvement, and more sophisticated performance prediction models to characterize real-time computing node behaviors for more accurate activity execution time estimation. It will be also of our interest to deploy a large network testbed to test large problem sizes in real environments.

Acknowledgment

This work is partially supported by U.S. National Science Foundation under Grant No. CNS-0721980 with University of Memphis and the Defence Pre-Research Project of the “Eleventh Five-Year-Plan” of China under contract No.513060601 with XiDian University.

References

1. Afrati, F.N., Papadimitriou, C.H., Papageorgiou, G.: Scheduling DAGs to minimize time and communication. In: Reif, J.H. (ed.) AWOC 1988. LNCS, vol. 319, pp. 134–138. Springer, Heidelberg (1988)
2. Agarwalla, B., Ahmed, N., Hilley, D., Ramachandran, U.: Streamline: a scheduling heuristic for streaming application on the grid. In: The 13th Multimedia Computing and Networking Conf., San Jose, CA (2006)

3. Bao, L., Chen, P., Zhang, X.: Batch invocation of web services in BPEL process. In: Bouguet-taya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 511–516. Springer, Heidelberg (2008)
4. Bashir, A.F., Susarla, V., Vairavan, K.: A statistical study of the performance of a task scheduling algorithm. *IEEE Trans. on Computer* 32(12), 774–777 (1975)
5. Biskup, J., Carminati, B., Ferrari, E., Muller, F., Wortmann, S.: Towards secure execution orders for composite Web services. In: Proc. of the IEEE International Conference on Web Services, pp. 489–496 (2007)
6. Chafle, G., Chandra, S., Karnik, N., Mann, V., Nanda, M.G.: Improving performance of composite Web services over a wide area network. In: Proc. of the IEEE Congress on Services, pp. 292–299 (2007)
7. Chafle, G., Chandra, S., Mann, V., Nanda, M.G.: Decentralized orchestration of composite Web services. In: Proc. of ACM Int. Conference on World Wide Web (WWW 2004), May 17–22, pp. 134–143. ACM, New York (2004)
8. Chaudhary, V., Aggarwal, J.K.: A generalized scheme for mapping parallel algorithms. *IEEE Trans. on Parallel and Distributed Systems* 4(3), 328–346 (1993)
9. Chen, L., Agrawal, G.: Resource allocation in a middleware for streaming data. In: Proc. of the 2nd Workshop on Middleware for Grid Computing, Toronto, Canada (October 2004)
10. Chen, S., Bao, L., Chen, P.: OptBPEL: A tool for performance optimization of BPEL process. In: Pautasso, C., Tanter, É. (eds.) SC 2008. LNCS, vol. 4954, pp. 141–148. Springer, Heidelberg (2008)
11. A. Endpoints. Activebpel engine architecture (version 4.1) (2008), <http://www.activebpel.org/docs/architecture.html>
12. Ferrante, J., Ottenstein, K.J., Warren, J.D.: The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and System* 9(3), 319–349 (1992)
13. Foggia, P., Sansone, C., Vento, M.: A performance comparison of five algorithms for graph isomorphism. In: Proc. of 3rd IAPR-TC-15 Int. Workshop Graph-based Representations in Pattern Recognition, pp. 188–199 (2001)
14. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, New York (1979)
15. Gerasoulis, A., Yang, T.: A comparison of clustering heuristics for scheduling DAG's on multiprocessors. *J. of Parallel and Distributed Computing* 16(4), 276–291 (1992)
16. Guerin, R., Orda, A.: Computing shortest paths for any number of hops. *IEEE/ACM Trans. Networking* 10(5), 613–620 (2002)
17. Guo, H., Huai, J., Li, H., Deng, T., Li, Y., Du, Z.: ANGEL: optimal configuration for high available service composition. In: Proc. of IEEE Int. Conference on Web Services, July 2007, pp. 280–287 (2007)
18. Hopcroft, J., Wong, J.: Linear time algorithm for isomorphism of planar graphs. In: Proc. of the 6th Annual ACM Symp., Theory of Computing, pp. 172–184 (1974)
19. Jordan, D.: *Web services business process execution language version 2.0*. OASIS Specification (2007)
20. Kwok, Y.K., Ahmad, I.: Dynamic critical-path scheduling: an effective technique for allocating task graph to multiprocessors. *IEEE Trans. on Parallel and Distributed Systems* 7(5) (May 1996)
21. Kwok, Y.K., Ahmad, I.: Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys* 31(4), 406–471 (1999)
22. Li, W., Zhao, Z., Fang, J., Chen, K.: Execution optimization for composite services through multiple engines. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 594–605. Springer, Heidelberg (2007)

23. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. of Computer System Science*, 42–65 (1982)
24. Mann, V.: Symphony: decentralized orchestration of composite Web services (2007), <http://domino.research.ibm.com>
25. Nanda, M.G., Karnik, N.: Synchronization analysis for decentralizing composite Web services. In: *Proc. of ACM Symposium on Applied Computing (SAC 2003)*, Melbourne, Florida, USA. ACM, New York (2003)
26. Sekhar, A., Manoj, B.S., Murthy, C.S.R.: A state-space search approach for optimizing reliability and cost of execution in distributed sensor networks. In: *Proc. of Int. Workshop on Distributed Computing*, pp. 63–74 (2005)
27. Shin, K., Han, S.: Efficient Web services composition and optimization techniques. In: *Proc. of IEEE Int. Conference on Web Services*, July 2007, pp. 1160–1161 (2007)
28. Shirazi, B., Wang, M., Pathak, G.: Analysis and evaluation of heuristic methods for static scheduling. *J. of Parallel and Distributed Computing* (10), 222–232 (1990)
29. Wu, Q., Gu, Y.: Supporting distributed application workflows in heterogeneous computing environments. In: *Proc. of the 14th IEEE Int. Conf. on Parallel and Distributed Systems*, Melbourne, Australia, December 2008, pp. 3–10 (2008)
30. Yildiz, U., Godart, C.: Towards decentralized service orchestrations. In: *Proc. of the 2007 ACM Symposium on Applied Computing*, pp. 1662–1666 (2007)
31. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: Qos-aware middleware for Web services composition. *IEEE Tran. on Software Engineering* 30, 311–327 (2004)
32. Zhu, Y., Li, B.: Overlay network with linear capacity constraints. *IEEE Trans. on Parallel and Distributed Systems* 19, 159–173 (2008)

Optimal Service Selection Heuristics in Service Oriented Architectures

Emiliano Casalicchio¹, Daniel A. Menascé², Vinod Dubey³, and Luca Silvestri¹

¹ Dipartimento di Informatica Sistemi e Prod.
Università di Roma “Tor Vergata”, Roma, Italy
{casalicchio,silvestri}@ing.uniroma2.it

² Department of Computer Science,
George Mason University, Fairfax, VA, USA
menasce@gmu.edu

³ The Volgenau School of Information Technology and Engineering,
George Mason University, Fairfax, VA, USA
vdubey@gmu.edu

Abstract. Service Oriented Architectures allow service brokers to execute business processes composed of network-accessible loosely-coupled services offered by a multitude of service providers, at different Quality of Service (QoS) and cost levels. To optimize their revenue and the offered QoS level, service brokers need to solve the problem of finding the set of service providers that minimizes the total execution time of the business process subject to cost and execution time constraints. This optimization problem is clearly NP-hard. Optimized algorithms that find the optimal solution without having to explore the entire solution space have been proposed to solve problems of moderate size. A heuristic search of the sub-optimal solution scales to problems of large size and is appropriate for runtime service selection. This paper evaluates the performance of three heuristic service selection algorithms that are candidates for implementation in scalable service brokers. Our goal is to identify which algorithm provides the solution closest to the optimal and how many selections are evaluated to find the solution. The comparison is made over a wide range of parameters including the complexity of the business process topology and the the tightness of the QoS and cost constraints.

Keywords: Service Oriented Architecture, Web services, service composition, QoS, heuristics.

1 Introduction

The Service Oriented Architecture (SOA) model enables a market of services, where service providers (SPs) provide services at different QoS levels and at different cost. In this emerging market it makes sense to investigate mechanisms to properly select a set of services, characterized by different QoS and cost levels, that when composed together satisfy the QoS needs and cost constraints of the resulting business process (BP). This problem is referred in the literature as the

QoS-aware Service Selection or Optimal Service Selection problem [1,2,3,4,8,9,10,12,13,14].

The execution of a business process is coordinated by a service broker (or broker for short). The broker needs runtime and scalable mechanisms to solve the optimal service selection problem and to exploit the dynamics of the service marketplace characterized by potential and rapidly changing conditions in workload intensity, QoS level, and cost.

In [8], the authors provided a performance model that takes into account the business process structure, including cycles, parallel activities, and conditional branches, and computes the end-to-end execution time and cost for the business process. For some performance metrics (e.g., cost, availability, reputation) the composition is a trivial linear combination of the performance measure of the composing services. On the contrary, for other metrics such as execution time, we have a non linear function of the performance level of the services being composed. In that paper, we also provided an efficient algorithm, called JOSeS algorithm, that finds the optimal solution without resorting to an exhaustive search of the of solution space. This efficient optimal algorithm can only handle problems of small to moderate size. That paper also presented and thoroughly evaluated a heuristic solution that is compared here with other two proposed heuristics.

Several approaches can be used to solve the service selection problem. Current proposals use exact algorithms or heuristics (e.g., [2] or genetic algorithms [3]) to solve the QoS-aware (optimal) service selection problem for each request, whose exact solution has an exponential complexity. In [12], the authors define the problem as a multi-dimension multi-choice 0-1 knapsack one as well as a multi-constraint optimal path problem. A global planning approach to select an optimal execution plan by means of integer programming is used in [13]. In [1], the authors model the service composition as a mixed integer linear problem where both local and global constraints are taken into account. A linear programming formulation and flow-based approach is proposed in [4]. There, the authors consider not only sequential composition of services but also cycles and parallel activities. Algorithms for the selection of the most suitable candidate services to optimize the overall QoS of a composition are also discussed in [7]. A different approach, based on utility functions as the QoS measure, is used in [9], where the authors propose a service selection mechanism based on a predictive analytical queuing network performance model. Other contributions to the issue of service selection and composition can be found in [6,11].

This paper presents a performance comparison of runtime heuristic algorithms to evaluate their accuracy in finding the sub-optimal solution and their scalability to large size problems. The algorithms we consider conduct a heuristic search of the solution space in order to find a sub-optimal solution that is very close to the optimal solution but is obtained by examining a drastically reduced number of selections. In fact, the experimental studies reported in this paper show that the heuristic solutions come very close to the optimal solution (less than 9.6%

worse) after having examined a very small number of possible solutions (less than 9.45 on average versus 125,794 for the efficient optimal search).

The paper is organized as follows. Section 2 introduces the problem formulation. The optimal solution approach is described in Section 3. The heuristic solutions are described in Section 4. Experimental results are discussed in Section 5. Section 6 concludes the paper.

2 Problem Formulation

We use the average execution time of the business process (BP) as its main QoS metric. As previously discussed, this metric is a nonlinear function of the execution times of individual business activities and depends on the BP structure and composition constructs used. The extension to other performance metric is straightforward.

We assume that the probability density function (pdf) and cumulative distribution function (CDF) of the execution times of each SP are known. We also assume that the execution cost of each business activity provided by the SPs is given.

Let,

- A business process B be composed of N business activities $a_i, i = 1, \dots, N$.
- R_{max} be the maximum average execution time for B .
- C_{max} be the maximum cost for the execution of B .
- $R_{i,j}$ be the execution time for business activity a_i when implemented by service provider $s_{i_j} \in S_i$. $R_{i,j}$ is a random variable with a probability density function $p_{i,j}$ and a cumulative distribution function $P_{i,j}$.
- $C_{i,j}$ be the execution cost of business activity a_i when it is implemented by service provider $s_{i_j} \in S_i$.
- \mathcal{Z} be the set of all possible service provider selections of the business activities of B .
- $z \in \mathcal{Z}$ be a service selection of N service providers that support the execution of business process B .
- $z(k)$: service provider allocated to activity a_k in service selection z .
- $R(z)$ and $C(z)$ be the average execution time and the cost for associated with service selection z , respectively.

The Optimal Service Selection problem is formulated as a nonlinear programming optimization problem where the objective is to find a service selection z that minimizes the average execution time subject to cost constraints:

$$\begin{aligned}
 & \min R(z) \\
 & \text{subject to} \\
 & R(z) \leq R_{max} \\
 & C(z) \leq C_{max} \\
 & z \in \mathcal{Z}
 \end{aligned}$$

$R(z)$ is, in general, a complex nonlinear function that can be obtained from well known results from order statistics.

The Optimal Service Composition problem formulated above can be solved using two different approaches. The first is an optimal solution approach (Optimal Service Selection) that avoids doing an exhaustive search of the solution space \mathcal{Z} (e.g., the JOSeS algorithm proposed by the authors in [8]).

The second approach (Heuristic Service Selection) adopts a heuristic solution that reduces the problem complexity. In the following we compare the performances of three heuristics that scale to large size problems.

The first required step for both the optimal reduced search and the heuristic is to be able to extract from the BPEL code that describes the business process, an expression for the global average execution time and another for the total execution cost. This expression needs to take into account the structure of the business process as well as the execution times and cost of the individual business activities.

3 Optimal Service Selection

BPEL offers different constructs to combine business activities into a business process. The business logic is a structured activity obtained by putting together elementary business activities (in the following, the term business process and business logic are used alternatively). Each business activity is essentially a synchronous or asynchronous invocation of a Web service operation. The main construct a structured BPEL activity includes are: sequential control (`<sequence>`, `<switch>`, and `<while>`), non-deterministic choice (`<pick>`), and concurrency and synchronization of elementary activities (`<flow>`).

In [8], the authors showed how one can obtain an expression for the execution time R of a business process and its execution cost C directly from its structure described in BPEL or an equivalent tree-like representation. While the execution cost of a business process is the sum of the execution costs of the activities of the business process (plus eventually some additional overhead), the execution time depends on how the business activities are structured. For example, if we have a sequence of business activities a_1, \dots, a_n , and a service selection z , the execution time of the business process is $R(z) = \sum_{i=1, \dots, n} R_{i,j}$ where $s_{i,j}$ is the service provider assigned to activity a_i in z . The execution time of an activity a_i that is repeated n times and that is supported by service provider $s_{i,j}$ is simply $R(z) = n \times R_{i,j}$. In the case of deterministic or non deterministic choices, the computation of the total execution is easily computed as $R(z) = \sum_{i=1, \dots, n} q_i \times R_{i,j}$ where q_i is the probability that activity a_i is invoked. Finally, the execution time of the parallel execution of n business activities is given by $R(z) = \max_{i=1, \dots, n} \{R_{i,j}\}$.

The computation of the average execution time for a business process that has `<flow>` constructs is quite involved, especially in the case where execution times are random variables, and is described in [8].

Optimal service selection can be done in a naive way by enumerating all possible service selections and computing their execution time and cost. A more

efficient approach avoids generating selections such that their subselections already violate the execution time and cost constraint. Such an algorithm, called JOSeS algorithm, was presented in [8], and is used here for comparing the heuristic algorithms presented in the next section.

4 Heuristic Service Selection

We present two new heuristics—Fastest First (**FF**) and Cheapest First (**CF**)—and compare them with the high reduction in execution cost, low increase in execution time ratio (**hrClIR**) heuristic presented by the authors in [8]. The goal of the proposed heuristic solutions is to reduce the cost of finding the optimal solution, providing a sub-optimal selection as close as possible to the optimal.

Figure 1 shows the solution space and the feasible solution space of our problem. The solution space is the area delimited by the dashed lines, which indicate the lower and upper bounds for cost and execution time of the business process. The lower bound C_C for the execution cost is the cost obtained by selecting the cheapest service providers for each activity. Similarly, the upper bound R_S for the execution time can be obtained by selecting the slowest service provider for each business activity. On the contrary, the upper bound C_E for the execution cost is obtained by selecting the most expensive service provider for each activity, and the lower bound R_F for the execution time is obtained by selecting the fastest service provider for each activity. The feasible solution space is represented by the dotted area and is the the portion of the solution space delimited by the lower bound for execution time and execution cost and by the time and cost constraints (bold lines). We assume that the execution cost of a service provider is inversely proportional to its execution time.

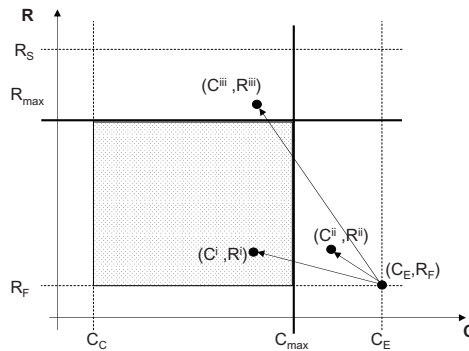


Fig. 1. A conceptual representation of the solution space and of the feasible solution space

4.1 Fastest First

The proposed heuristic is based on the following idea (see Fig. [10](#)). We start from the service selection z_0 characterized by the lowest execution time, i.e., the point (C_E, R_F) . Assume that this point is outside the feasible solution space and that the cost constraint is violated. To find a feasible solution as close as possible to the optimum, we have to choose a selection z' that moves the problem solution inside the dotted area, say the point (C', R') . To choose the solution z' , we determine the activity a_k such that the service provider allocated to a_k in z_0 , i.e., $z_0(k)$, provides the lowest average execution time among all allocated service providers in z_0 . We then replace $z_0(k)$ with the second fastest service provider for a_k .

We then evaluate the execution cost and execution time for the new service selection z' . If the constraints are satisfied we have a suboptimal solution (see point (C', R')). Otherwise, there are two possibilities: if the cost constraint is still violated and the time constraint is not yet violated, we are in a point such as (C'', R'') and we have to repeat the above mentioned process, i.e., the replacement of the fastest service provider for the new allocation z' .

If the execution time constraint is violated but the cost constraint is satisfied, we are at point (C''', R''') and we cannot accept such solution as we would continue to violate the execution time constraint at any further attempt of cost reduction. Then, we go back to the previous allocation (z_0 in this case) and we replace the service provider that has the second lowest execution time by its next fastest.

The details of this heuristic are shown in Algorithm [11](#). The function `GetFastest` (z, h) returns the service provider in allocation z that has the h -th smallest average execution time when $h \leq N$ and returns `NULL` when $h > N$. The function `next` (\mathbf{k}) returns the next, not yet evaluated, service provider in the list of service providers for activity a_k . This list is assumed to be sorted in increasing order of average execution time. This function returns `NULL` if all the providers for activity a_k have been already evaluated. We use the following notation in the algorithm. Let $z \ominus_k s$ stand for the operation of removing from solution z provider s for activity k . Similarly, let $z \oplus_k s$ denote the addition to solution z of provider s to activity k .

4.2 Cheapest First

The CF heuristic is based on the same criteria used in the Fastest First. The main difference is that the search for a sub-optimal solution starts from the point (C_C, R_S) in Fig. [11](#), which is the cheapest and slowest selection of service providers. Assume that this point is outside the feasible solution space and that the execution time constraint is violated. To find a feasible solution we have to choose a selection z' that moves the problem solution inside the dotted area. To choose the solution z' , we determine the activity a_k such that the service provider allocated to a_k in z_0 , i.e., $z_0(k)$, provides the lowest cost among all allocated service providers in z_0 . We then replace $z_0(k)$ with the service provider with the second lowest for a_k .

Algorithm 1. Fastest First Algorithm Solution

```

1: function FFHeuristic()
2: Find  $z$  such that  $E[R(z)] = R_F$ ;
3: if ( $E[R(z)] \leq R_{\max}$ ) and ( $C(z) \leq C_{\max}$ ) then
4:   return  $z$ 
5: end if
6: while  $C(z) > C_{\max}$  do
7:    $h \leftarrow 1$ ;
8:   while ( $s_{k_i} \leftarrow \text{GetFastestSP}(z, h) \neq \text{NULL}$ ) do
9:     if ( $s_{k_j} \leftarrow \text{next}(k) \neq \text{NULL}$ ) then
10:       $z \leftarrow z \ominus_k s_{k_i}$ ;
11:       $z \leftarrow z \oplus_k s_{k_j}$ ;
12:      if  $C(z) \leq C_{\max}$  then
13:        if  $E[R(z)] \leq R_{\max}$  then
14:          return  $z$ ;
15:        else
16:           $z \leftarrow z \ominus_k s_{k_j}$ ;
17:           $z \leftarrow z \oplus_k s_{k_i}$ ;
18:           $h \leftarrow h + 1$ ;
19:        end if
20:      end if
21:    else
22:       $h \leftarrow h + 1$ ;
23:    end if
24:  end while
25: end while
26: return infeasible solution
27: end function

```

We then evaluate the execution cost and execution time for the new service selection z' . If the constraints are satisfied we have a suboptimal solution. Otherwise, there are two possibilities: if the execution time constraint is still violated and the cost constraint is not violated, we have to repeat the above mentioned process, i.e., the replacement of the cheapest service provider for the new allocation z' .

If the cost constraint is violated but the execution time is satisfied we cannot accept such solution as we would continue to violate the cost constraint at any further attempt of cost increase. Then, we go back to the previous allocation and replace the service provider that has the second lowest cost to be replaced by its next cheapest.

4.3 hrCliR Algorithm

This heuristic, proposed in [8], starts evaluating the service selection z_0 characterized by the lowest execution time, i.e., the point (C_E, R_F) . Assume that this point is outside the feasible solution space and that the cost constraint is violated. To find a feasible solution as close as possible to the optimum, we have to

choose a selection z' that moves the problem solution inside the dotted area, say the point (C', R') . To choose the solution z' we replace the service provider that provides the highest reduction in the execution cost C with the lowest increase in the execution time R . To determine such provider, we need to compute the ratio

$$\Delta_{i,j,j'} = p_i \times \frac{C_{i,j} - C_{i,j'}}{R_{i,j'} - R_{i,j}} \quad j' > j \quad (1)$$

for each activity a_i ($i = 1, \dots, N$). In Eq. (1), j represents the service provider allocated to activity a_i , j' represents an alternate service provider for a_i , and p_i is the probability that activity a_i is executed in the business process. This probability is a function of the structure of the business process and its branching probabilities. We then select the activity for which there is an alternate provider that maximizes the value of the ratio for all such ratios. More precisely,

$$(k, m) = \operatorname{argmax}_{i=1, \dots, N; j' \neq j} \{ \Delta_{i,j,j'} \}. \quad (2)$$

According to Eq. (2), the service provider m when replacing service provider j in activity k yields the maximum value for the ratios Δ .

We then evaluate the execution cost and execution time for the new service selection z' . If the constraints are satisfied we have a suboptimal solution. Otherwise, there are two possibilities: if the cost constraint is still violated and the time constraint is not yet violated, we are in point such as (C'', R'') and we have to repeat the above mentioned process, i.e., the selection of a new service provider that maximizes the ratio Δ among all activities. If the execution time constraint is violated but the cost constraint is satisfied, we are at point (C''', R''') and we cannot accept such solution as we would continue to violate the execution time constraint at any further attempt of cost reduction. Then, we select the service provider that has the second best ratio Δ . The process is repeated until a feasible solution is found.

5 Experiments

We implemented the heuristics and the optimal JOSeS algorithm [8] to conduct experiments aimed at evaluating the efficiency of the former. In particular, we wanted to: 1) determine how close the heuristics solution are to the optimal, 2) compare the number of points in the solution space examined by each algorithm, 3) compare the three heuristics, CF, FF and hrCliR, over a wide range of parameters including the complexity of the business process topology, the tightness of the response time and cost constraints, and the number of SPs per activity.

5.1 Description of the Experiments

The experimental methodology and metrics computed mirrors pretty closely what the authors did in [8]. Fifty business processes were randomly generated and the expression for the average response time and execution costs were computed

according to section 3. The process for the generation of business processes determined randomly when to generate sequences, flows, switches (and their switching probabilities) as well as the number of branches of flows and switches.

The number of activities for the randomly generated business processes varied from 6 to 10. The number of flows and switches in these business processes varied in the range zero to three and zero to two, respectively.

The experiments assumed that the execution time of each service provider s is exponentially distributed. The cost of obtaining an average execution time $E[R_s]$ from service provider s was assumed to be equal to $1/E[R_s]$. In other words, the cost decreases with the inverse of the average service time offered by a service provider.

For each experiment, the number of SPs per activity $nspa$ was the same for all activities and that number was varied as follows: 2, 3, 4, 5, 6, and 7.

The complexity $\mathcal{C}(B)$ of a business process B is defined as

$$\mathcal{C}(B) = \#activities + \#flows + \sum_{\forall \text{ switch } i} \text{fanout}_i \quad (3)$$

using an adapted version of the control flow complexity and other metrics discussed in [5]. After all business processes are generated, we compute for each a normalized complexity $\mathcal{C}'(B)$ as follows

$$\mathcal{C}'(B) = \frac{\mathcal{C}(B) - \min_{\forall s} \mathcal{C}(s)}{\max_{\forall s} \mathcal{C}(s) - \min_{\forall s} \mathcal{C}(s)}. \quad (4)$$

It can be easily seen that $0 \leq \mathcal{C}'(B) \leq 1$ for any business process B .

We then apply the k -means, with $k = 3$, clustering algorithm on all business processes using $|\mathcal{C}'(B) - \mathcal{C}'(q)|$ as the distance between business processes B and q . The business processes in the cluster with the smallest centroid are called *simple* business processes, the ones in the cluster with the largest centroid are called *complex*, and the remaining ones *medium* business processes. The performance of the heuristics are also compared along this dimension.

For a given business process p and for a given number of SPs per activity, we compute the coordinates of the feasibility region (C_C, R_F) , (C_E, R_F) , (C_C, R_S) , and (C_E, R_S) . We then compute three sets of values for the constraints R_{\max} and C_{\max} according to how tight they are. We call them *strict*, *medium*, and *relaxed* constraints, and their values are:

$$\begin{aligned} C_{\max} &= C_C + (C_E - C_C)/3 && \text{for tight,} \\ R_{\max} &= R_F + (R_S - R_F)/3 && \text{for tight,} \\ C_{\max} &= C_C + (C_E - C_C)/2 && \text{for medium,} \\ R_{\max} &= R_F + (R_S - R_F)/2 && \text{for medium,} \\ C_{\max} &= C_E - (C_E - C_C)/6 && \text{for relaxed, and} \\ R_{\max} &= R_S - (R_S - R_F)/6 && \text{for relaxed.} \end{aligned}$$

We also carried out a set of experiments to evaluate the scalability of the algorithms. The number of activities was set to 10 and $nspa$ ranged from 5 to 40.

For the complexity of the experiments the heuristics were evaluated only in the *medium* constraints scenario.

5.2 Results of the Experiments

The following metrics are used to evaluate the heuristic algorithms discussed here.

- ε_R : absolute relative percentage average execution time difference defined as

$$\varepsilon_R = 100 \times \frac{|R_h - R_o|}{R_o} \quad (5)$$

where R_h and R_o are the average execution times obtained using the heuristic and JOSeS algorithm, respectively.

- ε_C : absolute relative percentage average execution cost difference defined as

$$\varepsilon_C = 100 \times \frac{|C_h - C_o|}{C_o} \quad (6)$$

where C_h and C_o are the average execution costs obtained using the heuristic and JOSeS algorithm, respectively.

- δ : the percentage of not feasible solutions found, defined as

$$\delta = 1 - \frac{N_h}{N_o} \quad (7)$$

where N_o is the number of optimal solutions found by the JOSeS algorithm (equal to the number of experiments) and N_h is the number of sub-optimal solutions found by the heuristic algorithm. Note that while N_o is equal to the number of experiments, N_h could be less than N_o because heuristics are not able to find a solution for all the constraints combination.

In [8], after conducting an exhaustive ANOVA analysis, we showed that both ε_R and ε_C depend on *nspa* and on the business process complexity ($\mathcal{C}'(p)$). Therefore, in the experiments, we compare the performance of the proposed heuristics for different value of the number of SPs per activity and for the three different classes of the normalized business complexity.

Our results can be summarized as follows:

1. The Fastest First and hrCilR heuristic algorithms have a comparable behaviour with an absolute relative percentage average execution time difference ε_R less then $7.3\% \pm 3.3\%$ at a 95% confidence interval on average and less then $9.6\% \pm 4.2\%$ at worst. The value of ε_C is always less then $4.8\% \pm 1.7\%$ at a 95% confidence interval.
2. The Fastest First and hrCilR heuristic algorithms need a similar number of iterations to find the sub-optimal solution (9.15 ± 1.35 for hrCilR and 9.45 ± 0.98 for FF in the worst case, that is for *tight* constraints). This number is five orders of magnitude less then the optimal JOSeS's algorithm (125,794 iterations on average).

3. The Fastest First and hrCilR heuristic algorithms find a sub-optimal (or optimal in some cases) solution in 95.6 percent of the cases and in 97.8%, respectively. On the contrary, the cheapest first has a high percentage of not found solutions (17.7%). We should mention that the heuristics are not able to find the optimal solution only when the constraints are *tight*.
4. The Cheapest First heuristic algorithm shows very high values of ε_R and ε_C , regardless of the type of constraints, the number of SPs per activity and the BP complexity. For example, in the case of $nspa = 5$ and relaxed constraints $\varepsilon_R = 123\% \pm 15.6\%$ at a 95% confidence interval.
5. FF and hrCliR scale for a wide range of $nspa$ (from 5 to 40) and the number of iterations to find a sub-optimal solution was always less than 6 ± 0.66 at a 95% confidence level.

Figures 2 and 3 show the performance of the heuristics for the three types of constraints and for $nspa = 3$ and $nspa = 5$, respectively. The results show that while Fastest First and hrCilR heuristic have the same behaviour in terms of relative percentage average execution time difference, the Cheapest First heuristic shows a very high value of ε_R , regardless of the type of constraints. This behavior can be explained as follows. The CF heuristic, starting from the cheapest solution, tries to find a sub-optimal solution that has a lower cost. Therefore, the goal of the Cheapest First is opposite to the optimization problem defined by Equation 1. Usually, the solution determined by the CF has a cost lower than the optimum, and the high value for ε_C is due to values of C_h less than C_o . On the contrary, the sub-optimal execution time is significantly higher than the optimum. From the experiments, it emerges also that the performance of CF drastically improves for tight constraints. In this scenario, the number of feasible allocations is reduced and the distance between the solutions is very small; therefore the probability of finding a solution near the optimum is higher.

Figures 4 and 5 show the values of ε_R for different values of business process complexity. Also along this dimension, the trend is confirmed, i.e., FF and hrCilR achieve very similar performance and outperform the Cheapest First algorithm.

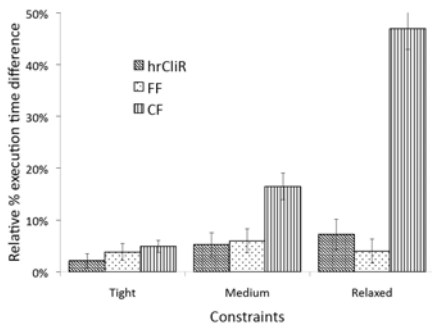


Fig. 2. ε_R as function of the type of constraints. In this scenario $nspa = 3$.

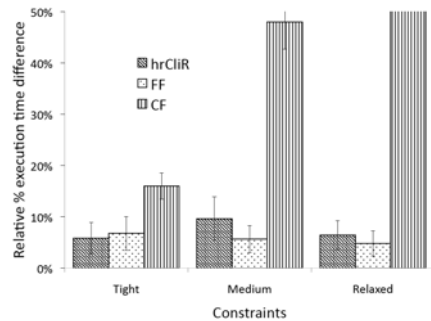


Fig. 3. ε_R as function of the type of constraints. In this scenario $nspa = 5$.

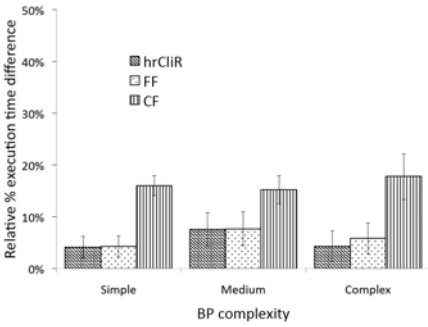


Fig. 4. ε_R as function of the business process complexity. In this scenario $nspa = 3$.

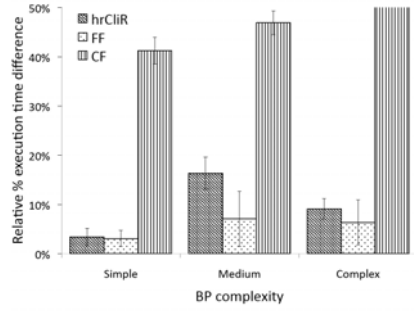


Fig. 5. ε_R as function of the business process complexity. In this scenario $nspa = 5$.

In the last set of experiments we evaluated the scalability of FF and hrCliR when the selection is done over a large set of SPs (from $nspa$ from 5 to 40) and for complex business processes. Figure 6 shows that the number of iterations to find a sub-optimal solution is always less than 6 ± 0.66 at a 95% at a confidence level. The hrCliR performs better than FF (at worst it takes 5.28 ± 0.54 iterations to find a solution). The weakness of this set of experiments is that it is impossible (with the systems we have access to) to compute the optimal solution and then the values of ε_R and ε_C . Therefore we are not certain of the goodness of the solutions found in the case of a large set of SPs.

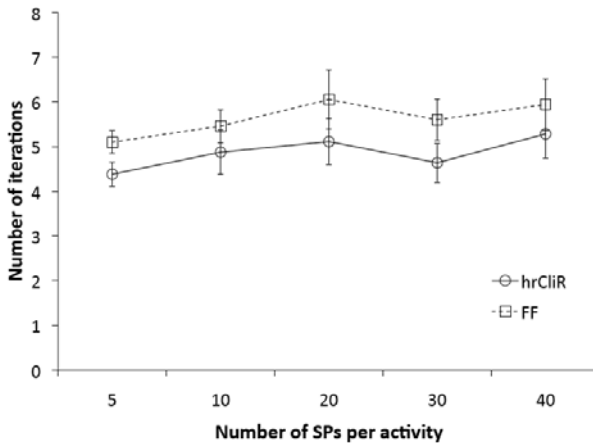


Fig. 6. Number of Iterations for medium constraints and complex BP

6 Concluding Remarks

The SOA model enables re-use and sharing of components through dynamic discovery. The benefit of service composition stimulates also the growth of a market of heterogeneous and volatile services. Service brokers are aware that: each possible service selection of services brings different levels of QoS and cost; and that the service marketplace environment is highly heterogeneous and volatile. Therefore, brokers need scalable mechanisms that can be used for runtime service selection among a set of service providers.

This paper presented two such efficient mechanisms that, in all experiments reported and all experiments carried out and not-reported due to lack of space, come very close to the optimal solution (less than 7.3% worse) after having examined a very small number of possible solutions (less than 9.45 worse).

Acknowledgment

The work of Daniel Menascé is partially supported by grant CCF-0820060 from the National Science Foundation.

The work of Emiliano Casalicchio and Luca Silvestri is partially supported by the PRIN project D-ASAP founded by the Italian Ministry of Education, University and Research.

References

1. Ardagna, D., Pernici, B.: Global and Local QoS Guarantee in Web Service Selection. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 32–46. Springer, Heidelberg (2006)
2. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: Heuristics for QoS-aware Web Service Composition. In: Proc. Int'l Conf. on Web Services (September 2006)
3. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: An Approach for QoS-aware Service Composition Based on Genetic Algorithms. In: Proc. Genetic and Computation Conf. (June 2005)
4. Cardellini, V., Casalicchio, E., Grassi, V., Francesco, L.P.: Flow-based service selection for web service composition supporting multiple qos classes. In: ICWS 2007. IEEE Intl. Conf. Web Services, July 9-13, pp. 743–750 (2007)
5. Cardoso, J., Mendling, J., Neumann, G., Reijers, H.A.: A Discourse on Complexity of Process Models. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 117–128. Springer, Heidelberg (2006)
6. Fung, C.K., Hung, P.C.K., Wang, G., Linger, R.C., Walton, G.H.: A Study of Service Composition with QoS Management. In: Proc. of the IEEE ICWS (2005)
7. Jaeger, M., Muhl, G., Golze, S.: Qos-aware composition of web services: A look at selection algorithm. In: Proc. 2005 IEEE Intl. Conf. Web Services, ICWS 2005 (2005)
8. Menascé, D.A., Casalicchio, E., Dubey, V.: On optimal service selection in Service Oriented Architectures. In: Performance Evaluation (in press)

9. Menascé, D.A., Dubey, V.: Utility-based QoS brokering in service oriented architectures. In: Proc. of the IEEE ICWS, Application Services and Industry Track, Salt Lake City, Utah, July 9-13, pp. 422–430 (2007)
10. Menascé, D.A., Ruan, H., Gomma, H.: QoS management in service oriented architectures. *Performance Evaluation Journal* 64(7-8), 646–663 (2007)
11. Serhani, M.A., Dssouli, R., Hafid, A., Sahraoui, H.: A QoS Broker based Architecture for Efficient Web Service Selection. In: Proc. 2005 IEEE ICWS (2005)
12. Yu, T., Lin, K.J.: Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In: Proc. of 3rd Int'l Conf. on Service Oriented Computing, December 2005, pp. 130–143 (2005)
13. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. *IEEE Trans. Softw. Eng.* 30(5), 311–327 (2004)
14. Cardellini, V., Casalicchio, E., Grassi, V., Lo Presti, F., Mirandola, R.: QoS driven runtime adaptation of service-oriented architectures. In: Proc. of the 7th ACM SIGSOFT ESEC/FSE 2009, Amsterdam, The Netherlands (August 2009)

Feedback-Based Adaptive Resource Control in QoS-Aware SOA Systems with Soft Real-Time Requirements

Francisco José Monaco and Pedro Northon Nobile

University of São Paulo,
Department of Computer Systems, ICMC
Av. Trab. Saocarlene, 400, São Carlos, SP,
CEP 13560-970, Cx.Po 668, Brazil
`monaco@icmc.usp.br`

Abstract. When deployed as operational components of production systems, novel computer services are supposed to respond synchronously to real-world events associated to the business process they implement, thereby needing to meet temporal constraints dictated by the dynamics of the environment in which they operate. This elicits a real-time system approach. One emerging concept to cope with such unpredictability of large-scale distributed computer applications is the use of feedback control principles. This paper introduces a feedback-based adaptive resource control algorithm for composite applications implementing real-time business process. The study is based on recent achievements in the field and ongoing progresses. A brief background on the field, the rationales of the proposed techniques and development results are presented.

Keywords: QoS, Real-Time, SOA, Control.

1 Introduction

Continuous technological advancements give rise to the dissemination of networked computer systems throughout our physical environment. From personal and house appliances like smartphones and digital assistants to mission-critical systems such as vehicular equipment and medical instrumentation, distributed computer application become commonplace in our ordinary routine. The more the applications supported by those systems are integrated into the implementation of on-line services upon which we rely for daily activities, the more relevant become their requisites of performance and dependability.

Unlike the asynchronous transactions characteristic of e-mail system, regular file transfer and other conventional Internet applications, those novel computer services are supposed to respond synchronously to real-world events associated to the business process they implement, and therefore need to meet temporal constraints dictated by the environment in which they operate. On this very need lies, in turn, the concept of real-time systems, whose specification poses

requirements on the expected response times, usually stated in the form of delay upper bounds. In this extent, when deployed as operational components of production systems, on-line services supporting security surveillance systems, computer-supported collaborative tools, mobile context-aware applications and even auctions management engines in an e-commerce system, to name a few examples, follow into this category, inasmuch as they need to timely respond to events that occur with respect to the natural (outer ambient) time, and thereby fulfill temporal responsiveness constraints.

In this field, an important paradigm gaining relevance in the design of large-scale distributed applications is that of service-oriented architecture (SOA). Based on the composition of complex application out of independently developed, loosely coupled, component services, the SOA approach relies on the combination of autonomous building blocks that interact to provide the desired functionality. This is achieved through the coordination of individual services to make up a composite service by means of asynchronous communication over an agreed implementation-independent protocol.

Most of the classical theory on real-time systems, nonetheless, arises from the domain of automation engineering, where the deterministic timing characteristics of industrial processes have made it possible the development of analytical techniques for the design and verification of mechanisms meant to operate under strict constraints. Contrasts with this scenario that of the typical infrastructure of today's interactive computer systems. The stochastic load patterns that applies to both the complex interactions of software and hardware resources in the system host, and the time-varying routing conditions of the network connecting them, yield a poorly predictable environment. In addition, unlike the typical periodic behavior of former automation systems, interactive computer services are inherently driven by event-oriented dynamics. Synchronization and fault-tolerance schemes in such asynchronous distributed systems become exceedingly more complex. As result, ensuring quality of service (QoS) in terms of performance requisites is considerably difficult in SOA systems. The extension of methodological and technical results from the real-time theory to address the non-deterministic features of interactive services is a research-demanding area [1].

The herein reported research work is aligned under this perspective focusing on the challenges of novel QoS-aware service architectures with performance requirements. This paper introduces a feedback-based adaptive resource control algorithm for large-scale composite applications implementing real-time business process. The study is based on recent achievements in the field and ongoing progresses. A brief background on the field, the rationales of the proposed techniques and development results comprise the subject of the next sections.

2 Background

In an intuitive sense, providing quality of service means fulfilling application requirements related to user-perceivable service effectiveness — being the user,

in SOA context, either a human or a system component implementing another service. A quality metric thus may be any performance or dependability parameter with meaningful value for the focused application such as throughput, latency, reliability or security requirements. Therefore, while requisites of real-time traffic for multimedia transmission on the Internet has been one major demand for the development of new resource allocation techniques and routing protocols for QoS provision at the network level, it is rather interesting that in an integrated mechanism the upper layers of the architecture cooperate with the policies implemented by the lower ones — an argument which has recently attracted attentions as a key feature for the deployment of effective computer services [2,3,4,5].

2.1 Challenges in Real-Time SOA

Among different parameters through which QoS can be formulated, responsiveness is a key-metric for interactive systems which gains increasing importance as the deployment of on-line services takes place in the control of business processes supported by the network infrastructure. More than influencing subjective user experience, responsiveness becomes a critical parameter when we move from time-independent data transfer to pervasive interactive applications. In this scenario, methodologies for design and evaluation of QoS-aware SOA elicits a real-time system approach.

An application designed with the assumption that any deadline miss implies in a service failure is known as a hard real-time (hard-RT) system. Associated to temporally non-deterministic infrastructures and workload conditions, however, typical large-scale SOA implementations do not lend themselves to such hard response-time requirements at a viable cost, unless under unrealistic worst-case assumptions. On the other hand, if a limited fraction of deadline misses does not imply in failure but, instead, in service degradation, the hard-RT specification can be relaxed. Real-time systems that can tolerate sporadic deadline misses have gained increasing importance in a context where the capability of meeting temporal requirements is associated to the QoS concept.

For this class of applications, stochastic responsiveness constraints are not only more technically and economically feasible, but also effectively meaningful regarding the needs of a wide range of application.

2.2 Average Performance Guarantees

Concerning services with non-rigid real-time requisites, the conventional metric to assess the delivered quality is the deadline miss ratio (DMR). In its simplest form, the dependability requirement may be stated as a DMR upper bound, which suffices to quantify the system reliability with respect to service failure rate, as well as the efficiency (effective throughput) regarding reissued requests (e.g. package retransmissions in a reliable communication mechanism). It does not, however, measure the distribution of delays in time. For the cases in which this is relevant (as in audio streaming of packet-switched networks),

DMR has been extended to consider either fixed [6] or sliding windows [7]. The (m, k) – *firm* [8], the *skip factor* [9] constraints comprise known examples. A more elaborated alternative based on Markov chains has also been recently introduced [10].

Those metrics are suited to firm real-time (firm-RT) systems, namely those in which requests whose deadlines are missed become useless, and are thus discarded (e.g. over-delayed live multimedia frames that should not be played if overdue). On the other hand, if tardy requests have degraded — rather than null — utility, and therefore should not be discarded, soft real-time (soft-RT) constraints are said to apply. This is a plausible scenarios for real-time SOA applications where, for a good consumer experience, the service delivering is expected to exhibit an upper-bounded average response time. It may sporadically take longer than that, but not so frequently that costumers need to wait for too long before being served. One subtle difference in this case is that consumers are not “droppable”; they just queue up in a line and remain there for their turn.

For soft-RT systems, a service-level agreement (SLA) based on DMR, or even windowed DMR, is not a key metric. Surely one wants to know how often unsatisfied clients will eventually timeout or give up from being served. Nonetheless, even if no one declines the transaction, user-perceivable quality of service can also be measured by how long costumers stand in the line.

An alternative metric which relates the service times to their occurrence frequency into a single measure is the average response time (ART) [11]. As a SLA parameter, the average response time (ART) can be meaningful in many circumstances where constraints on aggregate performance metrics are relevant. This is the case, for example, of an online soft-RT system with a finite buffer. It may not be necessary that the service consume queued requests at a constant rate implying in a hard-RT operation; in this case it suffices that the average response time be upper-bounded by a value which prevent the buffer from being emptied during the processes. This is valid for SOA systems whose semantic of the operations in the real-time business process implies soft-RT constraints. ART-based SLA models have been successful exploited in research works on SOA QoS provision.

2.3 Feedback Scheduling

A substantial deal of work in the area of real-time systems has produced important theoretical results aimed at the analysis of temporally deterministic applications. Processes characterized by periodic dynamics — commonly present in monitoring and control systems, as in industrial environment and vehicular automation — have motivated the development of sound analytical approaches. The treatment of event-driven (asynchronous) systems, where requests exhibit non-deterministic inter-arrival and execution time, is otherwise considerably more complex. Real-time resource allocation under non-deterministic dynamics is admittedly challenging even with state-of-art techniques. Heuristic approaches, instead, prevails in this domain.

Alternatively, one concept that has been emerging in the field of real-time computing is the *Feedback Control* notion, already established in other areas of Engineering. It grounds on the principle of self-adaptation by using the deviation of the system's output from the desired value as an input in such a way to force the system output contrarily to the deviation itself. This is called a negative closed loop, as illustrated in Figure 1.

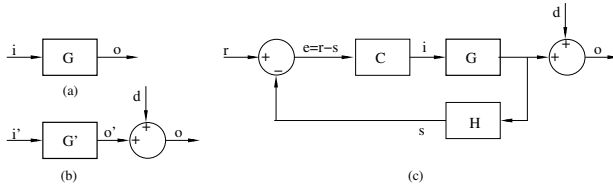


Fig. 1. Feedback control loop

In the diagram, a block represents a generic function $F(x(t))$ that converts the input signal $x(t)$ into an output signal $y(t)$. The controlled system is represented by the block G ; i and o denote the input and output respectively.

To set the output o at the desired value it is necessary to adjust the input i accordingly. In open loop control, Figure 1 (a), this has to be accomplished manually. If either some internal parameters of the system changes over time (changing G to G') or an external disturbance (d) affects the output, i must be re-adjusted to compensate for those changes. In the closed loop approach, Figure 1 (b), the block H is a sensor which measures the system output o and produce a corresponding signal s . By applying the control system a reference signal r equivalent to that assumed by s when the output has the desired value, the difference $e = r - s$ is the "error", or the instant deviation between the reference and the current output. This error signal is then injected into the controller block C so as to produce the signal $i = C(e)$, which is injected into the system G , forcing the output to the desired value. To see that, it suffices to suppose that the both controller and the sensor are proportional blocks $C(x) = P.x$ and $H(x) = Q.x$, so that $i = P.e$ and $s = Q.o$. If the system is currently delivering the correct output, then i must have the corresponding correct value. Any undesired increase in o , due to either a change in G or an external disturbance, will cause s to increase proportionally. This raise in s will, in turn, reflects into a decrease in e , since r is a constant reference signal. The system input i is also reduced, having as effect a corresponding decreasing in o . As it can be concluded, an increase in o , in the negative closed loop, forces a decrease in o . Conversely, a decrease in o , has the opposite effect. The system is always driven towards an output value that corresponds to the reference signal, and is thus much less susceptible to both internal parameter variations and external disturbances. Functions other than the bare proportional gain can be used as the controller block in order to shape the system's output as desired, and these include integrals and derivatives of the error time-function.

Control Theory has developed as a groundbreaking field in Engineering and in some natural science branches, and counts on a rich collection of mathematical modeling tools to describe the behavior of dynamic systems in terms of how they respond to different stimuli, and to verify stability, observability and controllability properties. It offers an extensive background for the design of control strategies applied to linear and non-linear systems.

The application of Control Theory in computer systems architectures for either performance or dependability optimization is nevertheless recent, and considerably less explored than in other domains. It has been employed, for instance, for dynamical clusterization of parallel processors in resource partitioning problems, and for throughput shaping in congestion control mechanisms. *Feedback scheduling* is considered a leading paradigm in real-time systems field [12,13,14,15], specially in applications meant for non-deterministic environments. In this case, runtime automatic parameter-tuning capability represents an advantageous alternative to the off-line presetting at design phase which may be only possible under over-pessimistic worst-case assumptions.

In the context of large-scale service-oriented architectures, self-adaptation constitutes an appealing concept. The simultaneous fulfillment of both the functional requirements concerning the overall business logic and the non-functional requirements concerning the QoS levels expected from the SOA system is a challenging goal in view of the stochastic dynamic of networked computer systems. Automatic control is a theoretically-grounded, well-established technique to cope with such unpredictability by means of closed-loop feedback, which has proven effective for building systems less susceptible to both outer and inner disturbances in other fields of Engineering. The exploitation of control theory principles in the design of self-adaptive QoS-aware SOA applications is a key approach towards the fulfillment of non-functional requirements in large distributed systems.

2.4 Scheduling for ART-Constraints

One important element influencing the performance of interactive services is the scheduler. In order to meet real-time constraints, appropriate scheduling policies should be applied to conveniently manage resource allocation.

For a set of service classes, let the goal be that of ensuring that the average response time calculated over a window of w past requests delivered to each system client be upper-bounded by a value agreed on a per-class basis. Intuitively, if one particular client is recurrently left aside in several consecutive scheduling cycles, the effective ART offered to it will tend to increase. One sensible approach is then to take the difference between both contracted and effective ART into consideration while assigning priorities to the resource allocation.

Classic results from the real-time theory have produced well-known scheduling algorithms [13], among which, the *earliest deadline first* (EDF) is an outstanding contribution. EDF is an optimum scheduling discipline for non-preemptive uniprocessors that assigns the highest priorities to the jobs with the shortest deadlines. It might seem that this urgency-based heuristic is a straightforward

solution for the highlighted problem, and that serving first those clients whose contracts are closer to a violation is reasonable.

A more careful theoretical exam under the light of the control theory, nevertheless, will clearly reveal that this is not the case. If the difference between contracted and effective ART is injected into the system as a feedback signal, and is the only factor influencing the scheduling decision, then this difference (the error) will be minimized. This means that the contracts will be always on the limit (a well-served client will be overlooked until it becomes bad served).

The *exigency-based scheduling policy* [16] is a newly proposed real-time scheduling algorithm developed in the scope of a research project whose aim is to enable the provision of absolute QoS guarantees in ART-constrained soft-RT systems. EBS assigns the highest priorities to the jobs with the lowest product given by Eq. (II)

$$P = D.C \quad (1)$$

where D is the deadline and C is the expected execution time. Pondering the urgency of a request with the execution time borrows the rationale of the *shortest job first* algorithm. SJF prioritizes the jobs with lower processing times and is known to minimize the average response time.

The EBS rationale is then to prioritize jobs with approaching deadlines but only if they are not too time costly. Recent works have shown the property of the EBS in delivering a fair balance of resource allocation proportional to the demands imposed by each service class [16].

3 Adaptive Resource Scheduling

Figure 2 shows a feedback loop depicting the adaptive resource scheduling architecture developed in this research work. The system input is the vector $[\mu_i]_n^1$ representing the ART upper bounds $\mu_i, i = 1, 2, \dots, n$ specified by each of the n clients. The output is the corresponding vector $[m_i]_n^1$ with the effective delivery ARTs.

The workload is represented by a queue of pending requests, where each job is parameterized by both the arrival time T and the cost C (execution time).

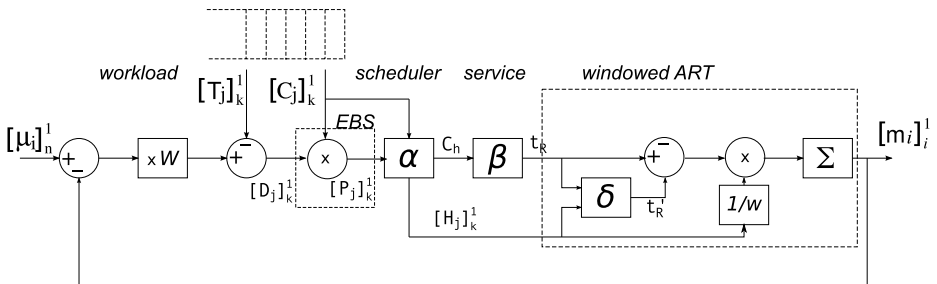


Fig. 2. The feedback loop of the adaptive resource allocation algorithm

Vectors $[T_j]_k^1$ and $[C_j]_k^1$ denote those two parameters, respectively, for all the k jobs in the queue.

If the job j currently in the queue was issued by the client i , then Eq. 2 represents the ART constraint:

$$\frac{m_i.w + T_j + D_j}{w} \leq \mu_i \tag{2}$$

The right side of the expression encompasses the total service time client i has experienced so far, plus the time the job has already spent in the queue, plus the time it will still wait if not selected now. The condition says that the average response effectively delivered should not be superior to the contracted value.

The maximum value D_j may assume in this inequation is the deadline for execution of the job before the contract is violated. Eq. 3 explicits this value:

$$D_j = (\mu_i - m_i).w - T_j \tag{3}$$

Notice that D_j is a then a time-varying deadline that is valid in every scheduling cycle and must be dynamically recalculated online.

Back to Figure 2, the EBS block receives the jobs' deadline and processing time and produces the priority vector. The block α represents the scheduler. It's function is to get the index h of the larges value in $[P_j]_k^1$ and then pick up the corresponding job in the queue. The other output of this block is a vector $[H_j]_k^1$ filled with zeros in all positions but h , where it holds the number 1.

The cost of the selected job is the input to the block β standing for the service, which is expected to exhibit a response time t_R .

The windowed upper-bounded ART constraint is depicted in the rightmost large block. The new average response time of the served client should be recalculate. An efficient way to perform this step is to calculate the new ART at a given instant from the ART at instant before by adding the value entering the window and subtracting the value leaving it at the other end, w positions back, as in Eq. 4:

$$m_z = \frac{m_z * w + t_z - t_{z-w}}{m} \tag{4}$$

Block δ is a FIFO of w positions which buffers incoming values of response time and returns the oldest instance in the window. The difference between current and last response time is multiplied by $1/w$ and summed up with the current effective ART, for the served client, to produce the output.

The result of this feedback architecture is an adaptive resource allocation which schedules the access to the service proportionally to the demands (effective service level and workload) of each client at every moment.

4 Development Results

The graphics in Figure 3 show the results of a simulation experiment for a system with two service classes A and B , between which the clients are distributed in

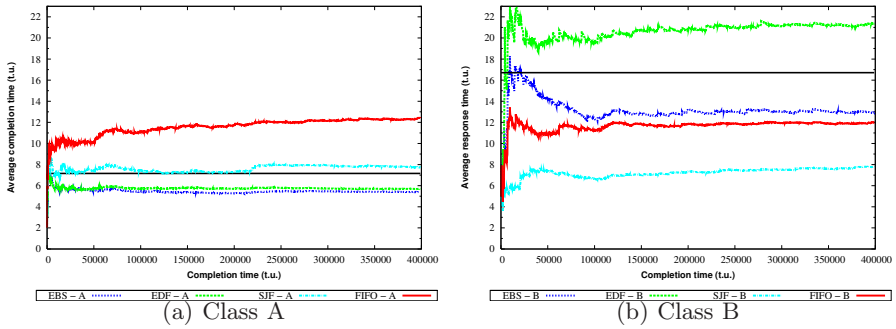


Fig. 3. Simulation results for the proposed architecture

the proportion of 1 : 3. The system is modeled as standard queue of independent jobs and a single non-preemptive processor.

The experiment was performed by means of a simulation program written in SMPL [17] discrete-event simulation library, which implements several queue scheduling algorithms.

A preliminary essay was run for a FIFO scheduler and the ART delivered to the clients was plotted. The horizontal axis represents the instants at which the service releases each job after having completely processed it, and the vertical axis denotes the ART calculated up to that point within a 200-job window. The graphic of Figure 3(a) refers to one client of class A, while that of Figure 3(b) is for one client of class B¹. The x-axis represent the time at which a request is released by the service after being completely processed, and the y-axis represents the average response time calculated at that moment.

The QoS contract was then established so that A-class clients are supposed to be assured an ART 40% below that provided by the FIFO scheduler, while B-class clients' ART upper bound is allowed to be 40% higher than that value. The horizontal line marks the contract levels of each service class. As it can be inferred, EDF heuristic performed disappointingly bad. That is because the scheduler directs all efforts to the clients whose contracts are near their limits, leaving aside the well-served ones until they also eventually approach their ART upper bound. The contracts are always at the limit and it is difficult to handle the stochastic load variations.

In the same graphics it is possible to see the simulation results for a *shortest job first* (SJF) scheduler. SJF prioritizes the jobs with lower processing times and is known to minimize the average response time. It is superior to EDF in this case but, unaware of any QoS contract, it performs equally good for both classes. It assigns more priority than need to fulfill contract B at the expenses of the more exigent contract A.

As it can be seen, out of the four essayed techniques, the proposed method (curve EBS in the graphic) is the only one capable of fulfilling both contracts,

¹ Service quality delivering is homogeneous for all the clients in a service class.

and it does so because it is able to loose in performance for class B , in order to decrease the effective ART delivery to class A . The constraint miss ratio calculated for this simulation was 98% for both classes, while the other algorithms achieved less than 40% for either A or B .

5 Conclusions

This paper introduces an adaptive resource control technique for QoS-aware distributed systems with soft real-time requirements.

The proposed architecture is modeled as a feedback loop and its operation rationales are explained. The research is motivated in the context of concurrent SOA implementing real-time business processes and investigates techniques aimed at request scheduling in concurrent services. Results of simulation experiments are used to discuss the properties of the algorithm with respect to the balance of the service scheduling among clients proportionally to the workload demands.

The present work explores the introduced formalization to describe recent results on QoS provision for ART-constrained systems, which are applicable in a wide varied of emerging network applications, specially those operating over communication infrastructures with stochastic performance. The practical use of the QoS model in the definition of novel SLA frameworks for network applications with responsiveness guarantees was addressed. It was also shown how the presented model can be explored in the design of scheduling strategies for resource allocation in computing and communication applications. The introduced adaptive resource scheduling architecture implements an efficient scheduling policy specially designed for ART-constrained real-time systems and represents an effective possibility for novel QoS approaches, and new business models for Internet service provision.

Related and ongoing works at the research group developing this work include and an extension of those results for heterogeneous multiprocessor systems, proposing a novel load balancing algorithm specially tailored for ART-constrained applications [18]. Another work introduces an adaptive admission control algorithm that protects the systems from overload by penalizing the QoS delivered to each client proportionally to the generated workload. A related project is devoted to the implementation of the proposed adaptive architecture as an extension for the mainstream Apache Web server, aimed at enabling service differentiation and performance guarantees in distributed Web architectures and SOA systems. Future works shall address other workload models such as heavy-tailed distributions and real-world.

Acknowledgments

Then authors thanks FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), and CAPESP (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the financial support offered to this research project.

References

1. Sha, L., Abdelzaher, T., Arzen, K.E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., Mok, A.K.: Real time scheduling theory: A historical perspective. *Real-Time Syst.* 28(2-3), 101–155 (2004)
2. Woodside, M., Menasce, D.: Guest editors' introduction: Application-level qos. *IEEE Internet Computing* 10(10), 13–15 (2006)
3. Wu, X., Li, M., Wu, J.: Enhanced Demand-driven Service Differentiation Algorithm in Web Clusters. In: *Proceedings of the IEEE International Conference on e-Business Engineering*, pp. 386–391 (2006)
4. Lee, S., Lui, J., Yau, D.: A Proportional-Delay DiffServ-Enabled Web Server: Admission Control and Dynamic Adaptation. *IEEE Transactions on Parallel And Distributed Systems*, 385–400 (2004)
5. Kang, C., Park, K., Kim, S.: A Differentiated Service Mechanism Considering SLA for Heterogeneous Cluster Web Systems. In: *Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WC-CIA 2006. The Fourth IEEE Workshop on*, p. 6 (2006)
6. West, R., Poellabauer, C.: Analysis of a window-constrained scheduler for real-time and best-effort packet streams. In: *Real-Time Systems Symposium, 2000. Proceedings. The 21st IEEE*, pp. 239–248 (2000)
7. Bernat, G., Cayssials, R.: Guaranteed on-line weakly-hard real-time systems. In: *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, pp. 25–35 (2001)
8. Hamdaoui, M., Ramanathan, P.: A dynamic priority assignment technique for streams with (m, k) -firm deadlines. *IEEE Transactions on Computers* 44(12), 1443–1451 (1995)
9. Koren, G., Shasha, D.: Skip-over: Algorithms and complexity for overloaded systems that allow skips. In: *Proceedings of the 16th IEEE Real-Time Systems Symposium*, pp. 110–117 (1995)
10. Liu, D., Hu, X., Lemmon, M., Ling, Q.: Firm Real-Time System Scheduling Based on a Novel QoS Constraint. *IEEE Transactions on Computers*, 320–333 (2006)
11. Monaco, F.J., Mamani, E.L.C., Nery, M., Peixoto, M.L.: A novel qos modeling approach for soft real-time systems with performance guarantees. In: *High Performance Computing and Simulation Conference - HPCS, Leipzig, Germany (June 2009)* (to appear)
12. Lu, Y., Abdelzaher, T., Lu, C., Sha, L., Liu, X.: Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. In: *Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE*, May 27–30, pp. 208–217 (2003)
13. Sha, L., Abdelzaher, T., Arzen, K.E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., Mok, A.K.: Real time scheduling theory: A historical perspective. *Real-Time Syst.* 28(2-3), 101–155 (2004)
14. Abdelzaher, T., Stankovic, J., Lu, C., Zhang, R., Lu, Y.: Feedback performance control in software services (2003)
15. Lu, C., Abdelzaher, T.F., Stankovic, J.A., Son, S.H.: A feedback control approach for guaranteeing relative delays in web servers. In: *RTAS 2001: Proceedings of the Seventh Real-Time Technology and Applications Symposium (RTAS 2001)*, Washington, DC, USA, p. 51. *IEEE Computer Society, Los Alamitos* (2001)

16. Casagrande, L.S., Monaco, F.J., de Mello, R.F., Bertagna, R., Filho, J.A.A.: Exigency-based real-time scheduling policy to provide absolute qos for web services. In: SBAC-PAD 2007: Proceedings of the 19th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2006), Gramado, RS, Brazil. IEEE Computer Society, Los Alamitos (2007)
17. MacDougall, M.H.: Simulating computer systems: techniques and tools. MIT Press, Cambridge (1987)
18. Monaco, F.J., Nery, M., Peixoto, M.L.: An orthogonal multi-resource real-time scheduling architecture for responsiveness qos requirements in soa environments. In: 24th Annual ACM Symposium on Applied Computing - ACM SAC, Honolulu, USA, March 8-12, pp. 1990–1995 (2009)

Author Index

- Acelas, Patricia 120
Alnasouri, Esam 218
Amorim, Marcelo Dias de 513
Ao, Antonio 647
Arce, Pau 120
Avallone, Stefano 545
Aznar, José I. 3
- Banchs, Albert 451
Bao, Liang 770
Bartolini, Novella 167
Belbekkouche, Abdeltouab 104
Benyamina, Djohara 86
Bezahaf, Mehdi 513
Bhattacharya, Abhishek 52
Bilski, Tomasz 251
- Cachopo, João 755
Calamoneri, Tiziana 167
Canonico, Roberto 545
Carbone, Paris 383
Cardellini, Valeria 431
Carvalho, Nuno 755
Casalicchio, Emiliano 431, 785
Castellanos, Wilder 120
Cerroni, Walter 417
Chen, Jian-Jia 317
Chen, Min 584
Chen, Ping 770
Choi, Jihyuk 351
Choi, Munhwan 493
Choi, Sunghyun 493
Choi, Young-June 334
Chou, Chun-Ting 696
Chu, Xiaowen 34
Ciotti, Roberto 569
Couceiro, Maria 755
Cui, Yong 203
Cuzzocrea, Alfredo 613
Csizmar Dalal, Amy 20, 730
- Dai, Huichen 770
Das, Sajal K. 203, 232
Diab, Ali 218
Dohnal, Vlastislav 400
- Douligeris, Christos 476
Dubey, Vinod 785
- Ernst, Rolf 280
- Fdida, Serge 513
Fernández-Navajas, Julián 3
Frenzel, Thomas 218
- Gandía, Jesús Díaz 556
Gelenbe, Erol 717
Gendreau, Michel 86
Ghinea, G. 530
Giotsas, Vasileios 476
Gonzalez, Sergio 584
Grassi, Vincenzo 431
Gu, Yi 770
Guerri, Juan C. 120
- Hafid, Abdelhakim 86, 104
Hollick, Matthias 451, 556
Hsieh, Hung-Yun 679
Hsu, Chun-Wei 696
Hu, Chunyu 363
Hu, Yih-Chun 351
- Iannone, Luigi 513
Ivers, Matthias 280
- Jia, Wei 770
Jiang, Yixin 34
Julien, Christine 131, 232
- Kalogeraki, Vana 383
Kang, Kyungtae 351
Katsaros, Dimitrios 613
Kawaler, Emily 20
Kim, Hwangnam 363
Ko, Chun-Han 663
Kretschmer, M. 530
- Leung, Victor C.M. 584
Li, Zi 148
Lin, Chuang 34
Lin, Hsiao-Pu 679

- Liu, Wenyu 317
 Liu, Xue 317
 Lo Presti, Francesco 431
 Lu, Xukang 265

 Makki, Kia 297
 Mangione, Stefano 463
 Manolopoulos, Yannis 613
 Massini, Annalisa 167
 Menascé, Daniel A. 785
 Mirandola, Raffaella 431
 Miridakis, Nikolaos I. 476
 Mitschele-Thiel, Andreas 218
 Mogre, Parag S. 556
 Monaco, Francisco José 799
 Monti, Gabriele 417, 627
 Moors, Tim 71
 Moro, Gianluca 417, 627
 Moyna, Niall 598

 Nahrstedt, Klara 185
 Navarra, Alfredo 569
 Nguyen, Hoang 185
 Niephaus, C. 530
 Noack, Andreas 739
 Nobile, Pedro Northon 799

 Pan, Deng 52, 297
 Papadimitriou, Alexis 613
 Park, Eun-Chan 363
 Park, Kyung-Joon 351
 Piazza, Francesco Ivan Di 463
 Pinotti, Cristina M. 569
 Pissinou, Niki 297
 Prasad, Narayan 334

 Qiao, Daji 493

 Rajamani, Vasanth 131
 Ramilli, Marco 417
 Rangarajan, Sampath 334
 Rao, Lei 317
 Rao, Nageswara S.V. 265
 Ravelomanana, Vlady 569
 Rezgui, Jihene 104

 Rivas, Raoul 185
 Roantree, Mark 598
 Rodrigues, Luís 755
 Romano, Paolo 755
 Roy, Nirmalya 232
 Ruiz, José 3

 Saldaña, José M^a 3
 Serrano, Pablo 451
 Shi, Jie 598
 Silvestri, Luca 785
 Silvestri, Simone 167, 717
 Sorbelli, Francesco Betti 569
 Stasi, Giovanni Di 545
 Steinmetz, Ralf 556

 Tinnirello, Ilenia 463
 Tong, Bin 148
 Tsai, Jack W. 71
 Tucker, Sam 20

 Vergados, Dimitrios D. 476
 Viruete, Eduardo 3
 Voller, Luca 451

 Wang, Guiling 148
 Wang, Shengling 34, 203
 Wang, Zongmin 265
 Wei, Hung-Yu 663
 Whelan, Michael 598
 Wu, Jianping 203
 Wu, Qishi 265, 770
 Wu, Zhung-Han 647

 Xiao, Yanping 34, 203
 Xu, Ke 203

 Yang, Zhenyu 52, 297
 Yeh, Ping-Cheng 647
 Yu, Jeonggyun 493

 Zezula, Pavel 400
 Zhang, Wensheng 148
 Zhang, Yan 584